

California State University, San Bernardino

**CSUSB ScholarWorks**

---

Theses Digitization Project

John M. Pfau Library

---

2003

## Telephone directory web service

Hua Sun

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Databases and Information Systems Commons](#)

---

### Recommended Citation

Sun, Hua, "Telephone directory web service" (2003). *Theses Digitization Project*. 2421.  
<https://scholarworks.lib.csusb.edu/etd-project/2421>

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact [scholarworks@csusb.edu](mailto:scholarworks@csusb.edu).

TELEPHONE DIRECTORY WEB SERVICE

---

A Project  
Presented to the  
Faculty of  
California State University,  
San Bernardino

---

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science  
in  
Computer Science

---

by  
Hua Sun  
September 2003

TELEPHONE DIRECTORY WEB SERVICE

---

A Project  
Presented to the  
Faculty of  
California State University,  
San Bernardino

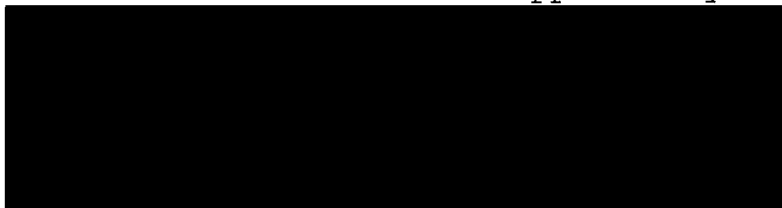
---

by

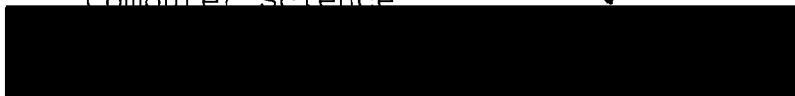
Hua Sun

September 2003

Approved by:



Dr. David Turner, Assistant Professor,  
Computer Science



Dr. Richard Botting



Dr. Tong Lai Yu

7/8/2003

Date

## ABSTRACT

Telephone Directory Web service (TDWS) is a Web service application that collects the telephone directories of different telephone companies. It provides a single point of access to the users. That is to say, TDWS has been designed for users to access telephone directory information aggregated from several telephone companies that expose their public data through a Web service interface.

TDWS consists of three main components: Information User, Information aggregator, and Information publisher with its database management system. A business entity providing a database of telephone numbers is named as an information publisher. The information aggregator, another business entity that maintains a telephone directory Web site, obtains data from the information publisher. An information user is a person accessing information through a Web browser. The information user sends Web requests for telephone numbers to the information aggregator. The information aggregator generates SOAP request messages after receiving the requests from the user's browser and passes SOAP requests to the information publisher. And then the information publisher extracts information from the database of telephone numbers using its own logic and sends SOAP responses as results to the information aggregator. Finally, the information aggregator formats HTML responses and forwards the HTML response to the information user.

TDWS is a simple web application consisting of complicated processes. The TDWS system was implemented through JavaServer Page, Java programming language, Hypertext Transfer Protocol, Simple Object Access Protocol, Extensible Markup Language, and other Web service technologies and non-proprietary standard protocols on Linux Operating System. Moreover, the TDWS software provides extensibility for further development.

Using TDWS, users can easily find telephone directory information through the Internet. TDWS demonstrates the application of Web services to the problem of providing convenient and cost-effective access to public telephone directory data. The advantage of the Web service model for the information publisher is that it can publish its information with lower cost. This is accomplished by replacing complicated presentation logic within the information aggregator's computing systems with a simple content-only Web service Application Program Interface (API). The advantage of the Web service model for the information aggregator is that it can eliminate the need to maintain local data stores and avoid the problem of serving stale data to the information user. The advantage of the Web service model for the information user is that information is more easily obtained through the information aggregator and the information is not stale.

## ACKNOWLEDGMENTS

I would like to thank California State University, San Bernardino, and the department of Computer Science for supporting me in finishing my master project.

I would like to thank Dr. David Turner, my advisor, who suggested the project and provided valuable suggestions and numerous discussions that vastly improved the quality of the project, who guided me from the very beginning of the project in the steps I needed to get it done.

I would also like to thank my committee members Dr. Richard Botting, and Dr. Tong Lai Yu for their valuable suggestions and comments.

I would like to extend my gratitude to Mr. Kwon Han and Scott who provided me useful suggestion and help when I set up the system, specially the SSH system.

Special gratitude goes to my family, particularly my mother Mrs. Li, PeiZhen who showed great understanding all along and stayed with me all the time.

Finally, I would like to thank the faculty and the staff of the Department of Computer Science, who made my stay pleasant in the school years.

## TABLE OF CONTENTS

ABSTRACT . . . . .	iii
ACKNOWLEDGMENTS . . . . .	v
LIST OF TABLES . . . . .	viii
LIST OF FIGURES . . . . .	ix
CHAPTER ONE: INTRODUCTION	
1.1 Purpose of Project . . . . .	1
1.2 Scope of Project . . . . .	2
1.3 Organization of Chapters . . . . .	3
CHAPTER TWO: TECHNOLOGIES	
2.1 Web Services . . . . .	5
2.1.1 Introduction to Web Services . . . . .	5
2.1.2 Web Service Technologies . . . . .	8
2.2 Network . . . . .	22
2.2.1 An Introduction to Network . . . . .	22
2.2.2 The Layers of Network . . . . .	22
CHAPTER THREE: PROJECT DESIGN	
3.1 Overall Description . . . . .	28
3.1.1 Project Perspective . . . . .	28
3.1.2 Project Functions . . . . .	31
3.2 Project Architecture Design . . . . .	32
3.2.1 The Architecture . . . . .	34
3.2.2 Description . . . . .	35
3.3 Information Aggregator Design . . . . .	37
3.3.1 Use Case Diagram . . . . .	37
3.3.2 Graphical User Interface . . . . .	38

3.4	Information Publisher Design . . . . .	42
3.4.1	Telephone Company One . . . . .	43
3.4.2	Telephone Company Two . . . . .	62
3.5	Advantages . . . . .	64
CHAPTER FOUR: PROJECT IMPLEMENTATION		
4.1	Project Classes Design . . . . .	66
4.1.1	Information Aggregator . . . . .	66
4.1.2	Information Publisher . . . . .	69
4.1.3	Database . . . . .	71
4.2	Project Classes Implementation . . . . .	75
4.2.1	Information Aggregator . . . . .	76
4.2.2	Information Publisher . . . . .	87
4.2.3	Database . . . . .	100
CHAPTER FIVE: FUTURE ENHANCEMENT AND CONCLUSION		
5.1	Future Enhancement . . . . .	104
5.2	Conclusion . . . . .	104
APPENDIX A: ACRONYMS AND ABBREVIATIONS . . . . .		107
APPENDIX B: SOURCE CODE . . . . .		111
REFERENCES . . . . .		211



## LIST OF TABLES

Table 3.1.	Software Interface . . . . .	29
Table 3.2.	City1 and City2 . . . . .	30
Table 3.3.	User Actions and Responses . . . . .	38
Table 3.4.	User Actions and Responses (Continue) . .	39
Table 3.5.	The Requirement of Customers' Information . . . . .	46
Table 3.6.	User Actions and Functions . . . . .	50
Table 4.1.	Information Aggregator Classes . . . . .	66
Table 4.2.	Information Aggregator Classes (Continue) . . . . .	67
Table 4.3.	Telephone Company One Classes . . . . .	69
Table 4.4.	Telephone Company Two Classes . . . . .	71
Table 4.5.	Person Table Design . . . . .	73
Table 4.6.	Phone Table Design . . . . .	74
Table 4.7.	Database Classes . . . . .	74

## LIST OF FIGURES

Figure 1.1.	Telephone Directory Web Service Integration . . . . .	1
Figure 2.1.	Two-tiered Model for Web Services . . . . .	5
Figure 2.2.	SOAP Request/Response Messages Model . . . . .	9
Figure 2.3.	A Request Message . . . . .	10
Figure 2.4.	A Response Message . . . . .	11
Figure 2.5.	The Client/Server Connection . . . . .	16
Figure 2.6.	Java for XML-based Remote Procedure Call Web Service . . . . .	20
Figure 2.7.	The Four-Layer Model of A Network . . . . .	23
Figure 2.8.	The Project Network . . . . .	25
Figure 3.1.	The Three-tiered Client/Server Architecture . . . . .	32
Figure 3.2.	Telephone Directory Web Service . . . . .	34
Figure 3.3.	Use Case Diagram . . . . .	38
Figure 3.4.	The Search Page . . . . .	40
Figure 3.5.	The Javadoc Page . . . . .	41
Figure 3.6.	The Codeviewer Page . . . . .	42
Figure 3.7.	Telephone Company One's Web Service . . . . .	43
Figure 3.8.	Relational Database Schema Diagram . . . . .	44
Figure 3.9.	Entity Relationships Diagram . . . . .	45
Figure 3.10.	Web Process of Insertion Function . . . . .	48
Figure 3.11.	Web Process of Deletion Function . . . . .	49
Figure 3.12.	Use Case Diagram of Administrators . . . . .	51

Figure 3.13. The Login Page . . . . .	51
Figure 3.14. The Menu Page . . . . .	52
Figure 3.15. Insert A New Record Page . . . . .	53
Figure 3.16. Insert Telephone Number Page . . . . .	54
Figure 3.17. The List Page of Insert Telephone Number . . . . .	55
Figure 3.18. The Delete Customer Page . . . . .	56
Figure 3.19. The List Page of Delete Customer . . . . .	57
Figure 3.20. Delete Customer's Phone Number Page . . . . .	58
Figure 3.21. List Page for Deleting Phone Number . . . . .	59
Figure 3.22. Edit Customer Page . . . . .	60
Figure 3.23. The Search Page . . . . .	61
Figure 3.24. The Review Page . . . . .	62
Figure 3.25. Telephone Company Two's Web Service . . . . .	63
Figure 4.1. Information Aggregator Class Diagram . . . . .	68
Figure 4.2. Telephone Company One . . . . .	70
Figure 4.3. Telephone Company Two Class Diagram . . . . .	72
Figure 4.4. Database Class Diagram . . . . .	75
Figure 4.5. The SearchPhoneNumber Class . . . . .	76
Figure 4.6. The TelcoClient Class . . . . .	77
Figure 4.7. The SocketTelcoClient Class . . . . .	78
Figure 4.8. The JAXRPCTelcoClient Class . . . . .	79
Figure 4.9. The Request Class . . . . .	80
Figure 4.10. The HTTPPartOfRequest Class . . . . .	81
Figure 4.11. The SOAPXMLPartOfRequest Class . . . . .	82

Figure 4.12. The Response Class . . . . .	83
Figure 4.13. The SOAPXMLResponseParser Class . . . . .	84
Figure 4.14. The MyPhoneBean Class . . . . .	85
Figure 4.15. The PhoneNumberItems Class . . . . .	86
Figure 4.16. The Telco1WebService Class . . . . .	87
Figure 4.17. The RequestHandler Class . . . . .	88
Figure 4.18. The RequestHandler Class (Continue) . . . . .	89
Figure 4.19. The SOAPXMLRequestParser Class . . . . .	90
Figure 4.20. The Response Class . . . . .	91
Figure 4.21. The HTTPPartOfResponse Class . . . . .	92
Figure 4.22. The SOAPXMLPartOfResponse Class . . . . .	93
Figure 4.23. The DBHandler Class . . . . .	94
Figure 4.24. The DBConnection Class . . . . .	95
Figure 4.25. The JAXRPCTelcoIF Class . . . . .	96
Figure 4.26. The JAXRPCTelcoImple Class . . . . .	97
Figure 4.27. The DBHandler Class . . . . .	98
Figure 4.28. The JAXRPCTelcoFetcher Class . . . . .	99
Figure 4.29. The CreateTelco Class . . . . .	100
Figure 4.30. The CreatTelco Class (Continue) . . . . .	101
Figure 4.31. The TelcoBean Class . . . . .	102
Figure 4.32. The TelcoBean Class (Continue) . . . . .	103

## CHAPTER ONE

### INTRODUCTION

#### 1.1 Purpose of Project

The purpose of this project is to design Telephone Directory Web Service (TDWS) that can be easily used to search telephone directory information aggregated from several telephone companies.

The operating flow chart of TDWS is shown in Figure 1.1. The information publisher with database is a Telephone Company that publishes it's database of telephone numbers as a Web service. It provides a Web service API so that other business entities may automate their access to this public information. The information aggregator, a business entity, maintains a telephone directory Web site developed using the web Service. It obtains data from the information publisher. An information user, who is a person, accesses information through a Web browser.

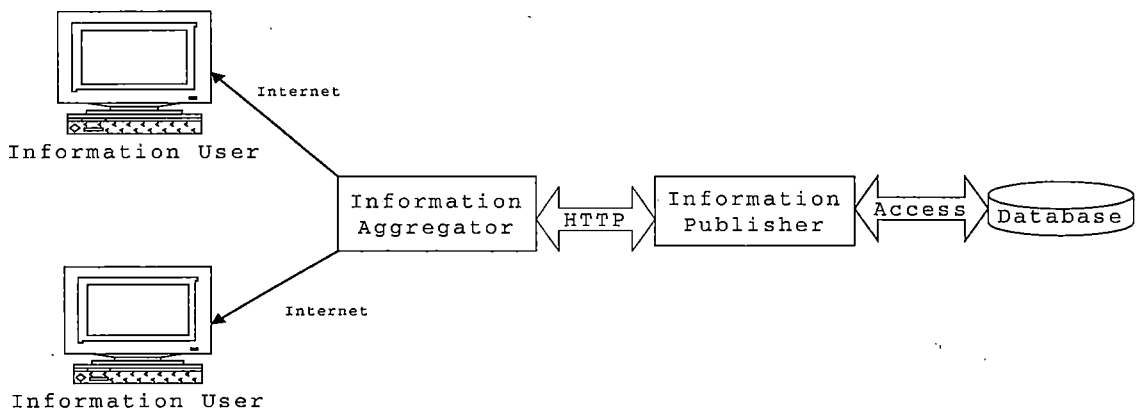


Figure 1.1. Telephone Directory Web Service Integration

## 1.2 Scope of Project

This project demonstrates and utilizes the application of Web services to provide convenient and cost-effective access to public telephone directory data. In order to realize this goal, two telephone companies, the information publisher, providing Web services with their separate telephone number databases are simulated. Another business entity (the information aggregator) with a Web site, whose sole purpose is to provide access to telephone directory data, has also been constructed. The information aggregator will provide a function to users, i.e., a data entry form for users to request telephone numbers of people they specify. When the information aggregator receives a Web request, it converts the Web request to a SOAP request message and transfers the SOAP request message to an appropriate telephone company. The telephone company then starts to search its own database to see if any stored information matches the request. If so, the telephone company sends a SOAP response message to the information aggregator. After obtaining the response message from the telephone company, the information aggregator formats a response message—including names, addresses, telephone numbers—with Hypertext Markup Language (HTML), which is a set of codes that determine how a Web page will appear, including graphics, links, and text characteristics.

Finally, the information aggregator presents it to the information user through the user's Web browser interface.

### 1.3 Organization of Chapters

Chapter Two introduces technologies used in this project. Web services are described with a brief overview of the associated technologies. Since this project is based on the network, the fundamental networking concepts are covered. Postgresql, which was used to develop database management system, is also presented

The architecture of this project is described in Chapter Three. Two telephone companies that provide Web services to their public information are described in detail. In this chapter, the definitions of the system components, interfaces, and specific requirements for the information publisher and the information aggregator are also provided. Graphical User Interface (GUI) of TDWS is provided in detail.

Chapter Four describes the implementation of the Telephone Directory Web Services.

Chapter Five presents future work and gives some conclusions of the project. Universal Discovery, Description and Integration (UDDI)[5] and security are briefly described as future developments.

An appendix provides definitions, acronyms, and abbreviations used in this project.

The Reference section provides some articles that provide knowledge sources for developing this project.



## CHAPTER TWO

### TECHNOLOGIES

#### 2.1 Web Services

Telephone Directory Web Service (TDWS) is a Web service application. In order to understand TDWS, it is necessary to introduce the definition of Web services and explain the technologies used to build Web services. The concept of networking is also described briefly.

##### 2.1.1 Introduction to Web Services

Web services are services offered via the Web. A Web service is a network accessible interface to application functionality, which is built using standard Internet technologies. The Web service architecture used in TDWS is shown in Figure 2.1. Components, Web service provider and Web service client, are presented in Figure 2.1.

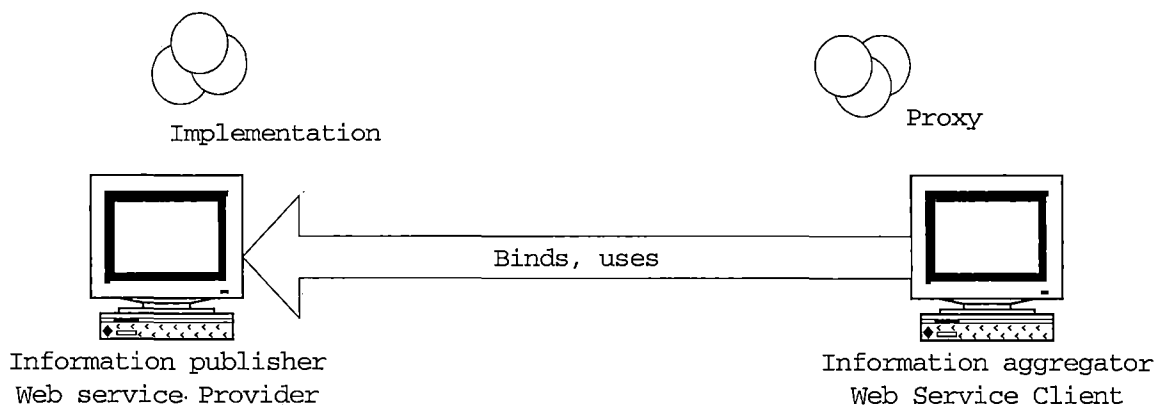


Figure 2.1. Two-tiered Model for Web Services

The Web service provider and the Web service client in Figure 2.1 are typical business entities, therefore Web services are predominantly business-to-business (B-to-B) transactions. An enterprise can be the provider of Web services and at the same time, can also be the client of other Web services. Web services don't need complex Web sites and graphical user interfaces. They are interfaces provided to users via the Internet, and they make a distributed environment.

To access Web services providers' database management system (DBMS), Web service clients do not need to go to Web service providers' Web sites or use Web service providers' complicated applications to access Web service providers' DBMS. Instead, the clients can use Web service provided by Web service providers to build their simple and creative applications that can be easily used. These applications send Simple Object Access Protocol (SOAP) [2] over Hypertext Transfer Protocol (HTTP) [1] requests directly to programs running on Web service providers' computing systems. The programs access Web service providers' DBMS and get needed information. Web service providers' DBMS is remotely accessed via Web services and the clients' applications.

In TDWS, the information publisher is a Web service provider and the information aggregator is a Web service client, as shown in Figure 2.1. For instance, the information aggregator develops an application based on a

Web service provided by the information publisher. The information user uses a Web browser to send a Web request for telephone numbers to the information aggregator. The information aggregator takes that Web request, converts it to a SOAP protocol over HTTP protocol request and passes it directly to programs running on the Web service. The programs access the information publisher's DBMS, and return responded information. The information user and the information publisher share interior information stored in the information publisher DBMS via the Web service and the information aggregator's Web service application. Moreover, information publishers can exchange information with each other. This will be the future work for this project. In this project, there are two different information publishers: one is "Telephone Company One" developed manually by using Socket paradigm with Extensible Markup Language (XML) [3], SOAP protocol, and HTTP protocol; and the other one is "Telephone Company Two" developed by using JAX-RPC technology with Java framework. Telephone Company One can be a Web service provider or a Web service client of Telephone Company Two. They can use Web services to connect their internal business systems and exchange information each other.

To communicate with each other, Web service applications need the ability of communication even if they use different computing platforms. Whatever language is

used and whatever operating system is used, one Web service application should easily communicate with another one. This ability can be accomplished by using standard communication protocols and Web service technologies: HTTP, SOAP, XML schema, Web Service Description Language (WSDL) [4], Socket paradigm, Java API for XML-based Remote Procedure Call (JAX-RPC) technology, and Java API for XML Processing (JAXP) technology. The detailed description on these technologies and protocols is given below.

### 2.1.2 Web Service Technologies

2.1.2.1 Request/Response Messages. In this project, request and response messages sent back and forth between applications are XML documents stored in SOAP envelope over HTTP protocol request/response messages. The diagram shown in Figure 2.2 illustrates a construction of SOAP messages. The construction consists of three components: XML document, SOAP protocol, and HTTP protocol. The following examples are used to illustrate SOAP messages.

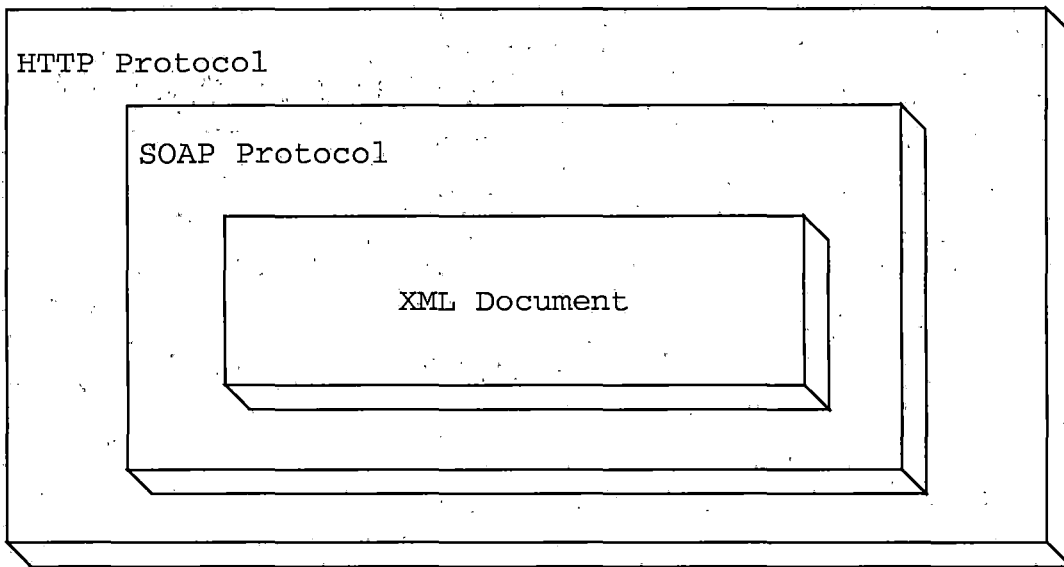


Figure 2.2. SOAP Request/Response Messages Model

2.1.2.1.1 Examples. Examples shown in Figure 2.3 and Figure 2.4 demonstrate SOAP request and response messages. These examples are SOAP messages used in this project. In the SOAP request examples, a RequestTelephoneNumber request is sent from the information aggregator to the information publisher. The request message has a personName parameter, and the ResponseTelephoneNumber response with a person parameter, which is shown in SOAP response example, is sent back to the information aggregator from the information publisher. The namespace used in SOAP messages is defined in "<http://tdws.ias.csusb.edu>" address, which is the host address of the machine on which the project was developed.

```

GET / HTTP/1.1
Host: tdws.ias.csusb.edu
Context-Type: text/soap+xml; charset="iso-8895-1"
Context-length: 602
SOAPAction: http://tdws.ias.csusb.edu

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
    <t:Transaction
      xmlns:t="tdws.ias.csusb.edu"
      SOAP-ENV:mustUnderstand="1">
        5
      </t:Transaction>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
      <m:RequestTelephoneNumber xmlns:m="http://tdws.ias.csusb.edu">
        <personName>
          <firstName></firstName>
          <middleName></middleName>
          <lastName>Sun</lastName>
          <location>city1</location>
        </personName>
      </m:RequestTelephoneNumber>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>

```

Figure 2.3. A Request Message

```

HTTP/1.1 200 OK
Context-length: 1319
Content-Type: application/soap+xml; charset="iso-8895-1"
Connection: close

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
    <t:Transaction
      xmlns:t="tdws.ias.csusb.edu"
      SOAP-ENV:mustUnderstand="1">
      5
    </t:Transaction>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <m:ResponseTelephoneNumber xmlns:m="http://tdws.ias.csusb.edu">
      <person>
        <size>2</size>
        <person1>
          <firstName>Bob</firstName>
          <middleName></middleName>
          <lastName>Sun</lastName>
          <street>223 West Second Street Apt. #1</street>
          <city>city1</city>
          <state>CA</state>
          <zipCode>92365</zipCode>
          <phoneNumber>(909) 369-5578</phoneNumber>
        </person1>
        <person2>
          <firstName>Bob</firstName>
          <middleName></middleName>
          <lastName>Sun</lastName>
          <street>223 West Second Street Apt. #1</street>
          <city>city1</city>
          <state>CA</state>
          <zipCode>92365</zipCode>
          <phoneNumber>(909) 778-0987</phoneNumber>
        </person2>
      </person>
    </m:ResponseTelephoneNumber>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Figure 2.4. A Response Message

2.1.2.1.2 Hypertext Transfer Protocol. HTTP is the standard protocol for communication between Web browsers and Web servers. HTTP specifies how the information aggregator and the information publisher establish a connection with each other, how the information aggregator requests telephone numbers from the information publisher, how the information publisher responds to that request, and finally how the connection is closed. In the above examples, there are some HTTP header fields:

- Request-Line The request line shows the method and HTTP protocol version. In the request example, Get method and HTTP 1.1 version are used. The Get method indicates that the request message retrieves whatever information with the specified person carried by the request message body.

- Host The Host indicates the Internet host name of resource being requested. The information origin in this project is the information publisher that is developed on the machine whose host name is "tdws.ias.csusb.edu". Therefore, "tdws.ias.csusb.edu" is in turn the host name of HTTP messages.

- Content-Type The content-type defines the type of message body. The content-type of SOAP messages used in this project is String data type.



- **Charset** The charset indicates what character sets are acceptable for the SOAP messages. In this project, iso-8895-1 is used.

- **Content-Length** The content-length indicates the number of bytes in the SOAP message body. The content-length of the example request is 602, and that of response example is 1319.

- **SOAPAction** Since the message body is XML in a SOAP envelope, the SOAPAction needs to be specified in HTTP request header fields. It indicates the intent of SOAP request.

- **Status-Line** In this response example, 200 and "OK" indicate the request for telephone numbers sent from the information aggregator has succeeded.

2.1.2.1.3 Simple Object Access Protocol. SOAP is a simple XML based protocol, which lets applications exchange information over HTTP. A SOAP message contains the following elements:

- **Envelope Element** The envelope element identifies XML documents as SOAP messages, and it is the root element of the XML document. The `xmlns:SOAP-ENV` namespace, <http://schema.xmlsoap.org/soap/envelope/>, defines the envelope as a SOAP envelope. The `soap:encodingStyle` attribute, <http://shemas.xmlsoap.org/soap/encoding/>,

indicates the data types used in the XML document, such as string, numeric, etc.

- Header Element Header element determines how the information aggregator and the information publisher process SOAP messages. The mustUnderstand global attribute indicates whether the header element is optional or mandatory. If the value of this attribute is "1", the information aggregator and the information publisher need to obey the semantics when they process the SOAP messages.

- Body Element In this project, the message body is XML document. The body element of the SOAP request message contains actual request information on the specified person's name and location, which is sent from the information aggregator to the information publisher. The body element of the SOAP response message contains actual response information with the specified person's telephone number and address, which is sent from the information publisher to the information aggregator.

2.1.2.1.4 Extensible Markup Language Part. XML is a cross-platform, software and hardware independent standard for transmitting information. XML is a kind of meta-language that can be read and understood by human and machines. XML describes information applied by both of the information aggregator and the information publisher.

2.1.2.2 Socket Paradigm. A socket is a connection interface between the information aggregator's application and the Web service of the information publisher. Socket paradigm is based on a client/server model. A client/server application stores large quantities of information on a server, while most of the program logic and graphical user interface is handled by client software. The Web is the most popular client/server system on the Internet. Web servers, such as Apache and Tomcat, store data and send responses to a Web client, such as Microsoft Internet Explorer. Typically, a client initiates a conversation, while a server waits for clients to start conversation with it. In this project, the socket paradigm is used to develop Telephone Company One's Web service and the application of the Web service. The construction of the client/server connection used in this project is shown in Figure 2.5.

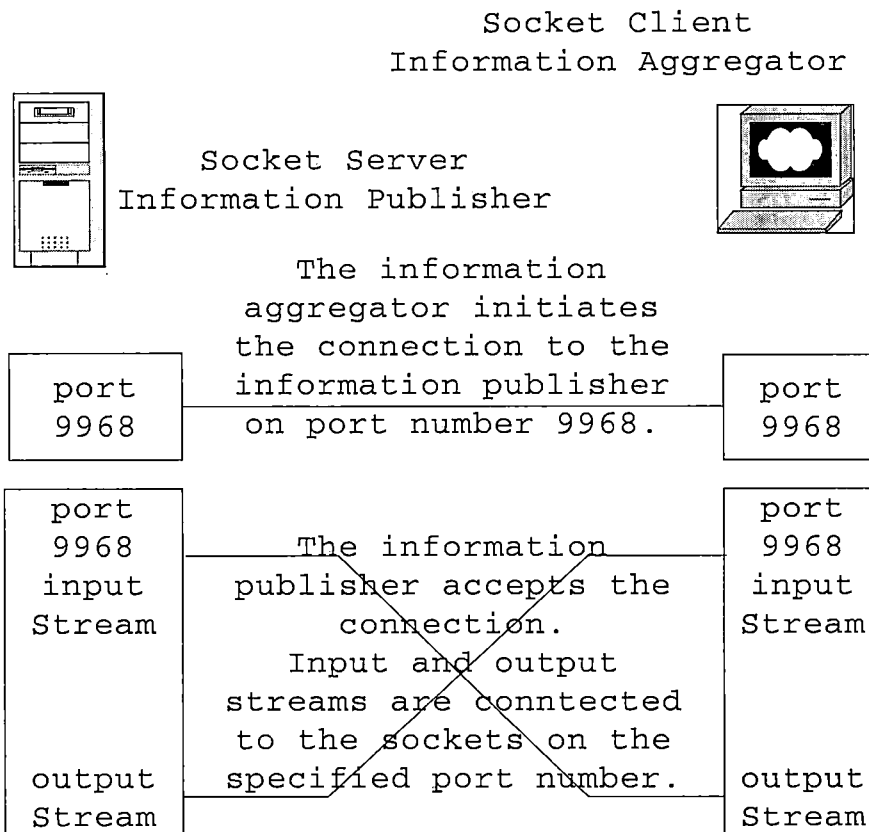


Figure 2.5. The Client/Server Connection

2.1.2.2.1 The Socket Client. In the Figure 2.5, the information aggregator can be treated as a socket client. The information publisher can be treated as a socket server. Normally, the information aggregator performs the following actions:

- The information aggregator creates a new socket.
- The socket attempts to connect to the remote host, i.e., the information publisher, with the IP address 139.182.137.21 and port number 9968.

- Once the connection is established, the information aggregator gets input and output streams from the socket and uses those streams to send request and receive response to and from the information publisher. This connection is full duplex; so both information aggregator and information publisher can send and receive SOAP messages simultaneously.

- The information aggregator receives Web requests from the information user, generates SOAP request messages, and sends them to the information publisher.

- After the information aggregator receives SOAP response messages from the information publisher, it parses XML documents that are located in SOAP envelope and gets actual response information. The information aggregator formats HTML response message with actual response information.

- The information aggregator sends HTML response to the information user.

- When the transmission of messages is complete, the information publisher breaks the connection.

2.1.2.2.2 The Socket Server. The information publisher performs the following actions:

- The information publisher creates a new server socket on a particular port number, 9968.

- The server socket listens to incoming connecting attempts on that port number from the information aggregator.

- The server socket gets input and output streams that communicate with the information aggregator.

- The information aggregator and information publisher interact according to an agreed-upon protocol, such as HTTP protocol and SOAP protocol, until it is time to close the connection.

- After the information publisher receives SOAP request messages from the information aggregator, it parses XML documents that is in SOAP envelope and gets actual request information.

- The information publisher connects postgresql server and accesses DBMS to look for information. After it gets response information, it formats XML document with response information in a SOAP envelope over HTTP protocol as a SOAP response message.

- The information publisher sends the SOAP response message to the information aggregator.

- The information publisher breaks the connection.

- The server socket waits for the next connection.

2.1.2.3 Java API for XML-based Remote Procedure Call Technology. JAX-RPC is an API for building and using Web services and Web services' applications. JAX-RPC Web

service implements procedures those are available for service customers to call. Since it is used in distributed client/server environment, an RPC mechanism allows Web service customers to execute procedures on other systems. To do this, JAX-RPC Web service needs to be deployed in a web container (such as Tomcat used in this project) on Web service providers' computing systems.

Using JAX-RPC technology with Java framework can develop JAX-RPC Web service. The JAX-RPC Web service also uses WSDL to describe itself, including its name, location, and procedures' names, etc. According to the WSDL, the Web service generates a stub as a local object for the Web service clients to present the remote JAX-RPC Web service and generates a tie as a proxy on the Web service providers to present the JAX-RPC Web service. Figure 2.6 shows the construction of JAX-RPC Web service and application. Telephone Company Two developed for this project is used as an example to illustrate how JAX-RPC Web service works.

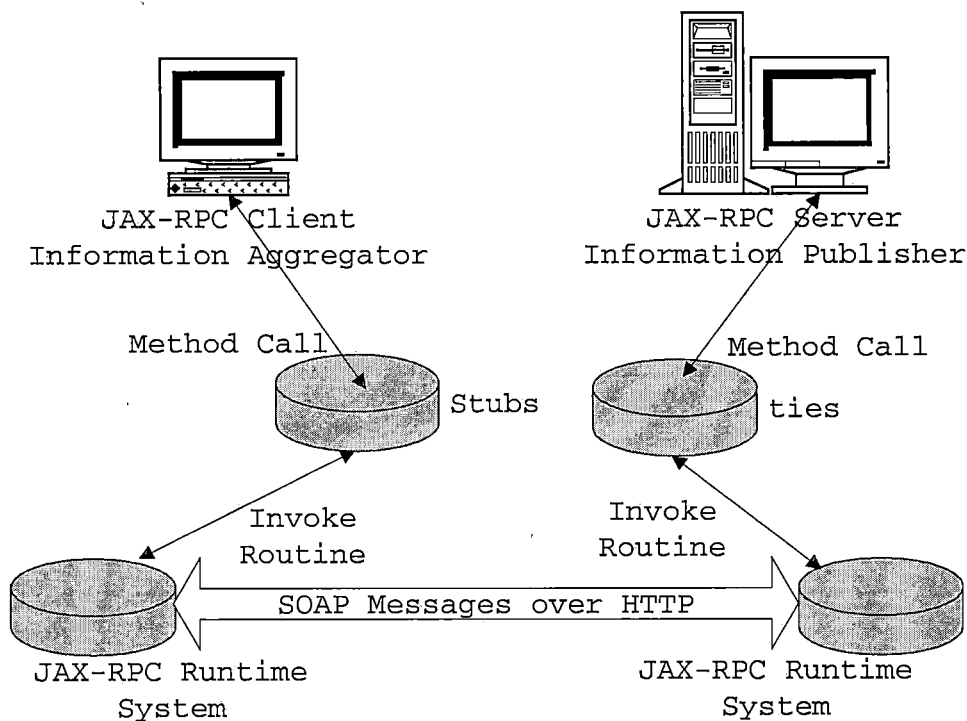


Figure 2.6. Java for XML-based Remote Procedure Call Web Service

In the diagram shown above, the information aggregator can be treated as a JAX-RPC client, and the information publisher can be treated as a JAX-RPC server. The JAX-RPC Web service life cycle is described as follows:

- To call a remote procedure, the information aggregator invokes a method on a stub and passes request information for telephone number as the method's parameters.
- The stub invokes routines in the information aggregator's JAX-RPC runtime system.
- The runtime system converts the remote method call into a SOAP message and then transmits the message as an HTTP request.



- When the information publisher receives the HTTP request, the information publisher's JAX-RPC runtime system extracts the SOAP message from the HTTP request and translates it into a method call.

- The JAX-RPC runtime system invokes the method on the tie object.

- The tie object invokes the method on the implementation of the information publisher's Web service.

- The runtime system on the information publisher converts the method's response into a SOAP message and then transmits the message back to the information aggregator as a HTTP response.

- On the information aggregator, the JAX-RPC runtime system extracts the SOAP message from the HTTP response and then translates it into a method response for the information aggregator.

2.1.2.4 Java API for XML Processing Technology and Others. This project uses Java API for XML Processing technology to generate a XML parser to parse SOAP messages. This project also uses Apache ANT, a Java-based build tool, to generate the information aggregator system and the information publisher systems separately.

2.1.2.5 Postgresql. In this project, postgresql is used to create the information publisher's DBMS. Postgresql is an object-relational database management system. Compared

to MySQL, postgresql has more functionality. For instance, postgresql supports foreign keys. The default setting for postgresql is to support transactions, whereas in mysql, transactions need to be turned on. In general, mysql is actually faster, but postgresql is more powerful. If an application just needs simple queries and does not have any need for large-scale update on tables that are being read, mysql is often faster than postgresql. If the application has a complex DBMS, postgresql would be a wiser choice.

## 2.2 Network

### 2.2.1 An Introduction to Network

A network is a collection of computers and other devices that can send and receive data to and from each other. A network is normally connected by wires, and the bits of data are turned into electromagnetic waves that move through the wires.

Sending data across a network is a complex operation. To make this complexity manageable, the different aspects of network communication are separated into multiple layers. The TCP/IP network model that is used in this project is presented in the next section.

### 2.2.2 The Layers of Network

Figure 2.7 shows the standard TCP/IP four-layer network model. In this model, applications, such as Netscape Navigator, run in the application layer and talk only to the

transport layer. The transport layer talks only to the application layer and the Internet layer. The Internet layer in turn talks only to the host-to-network layer and the transport layer, and it never directly talks to the application layer. The host-to-network layer moves the data across the wires, fiber-optic cables, or other medium to the host-to-network layer on the remote system, which then moves the data up the layers to the application on the remote system. The different layers with protocols are briefly described as Follows:

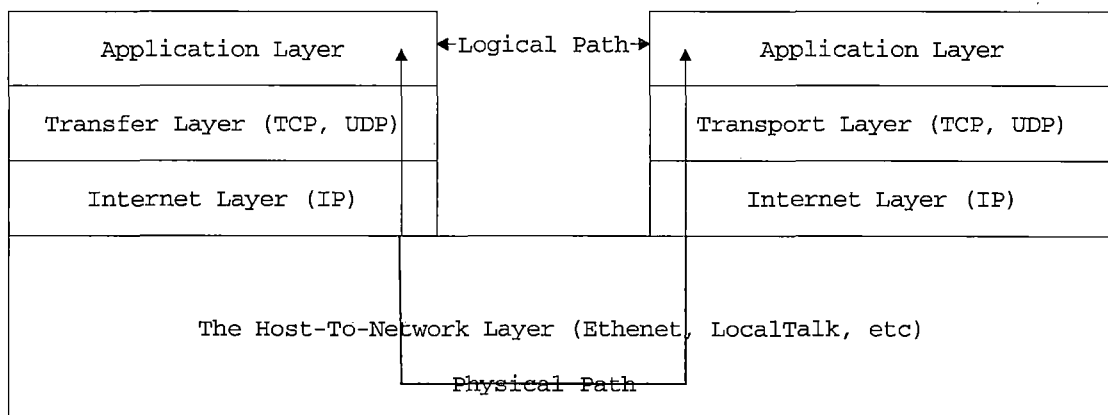


Figure 2.7. The Four-Layer Model of A Network

The host-to-network layer defines how a particular network interface sends IP datagrams (The data sent across the internet layer in packets are called as datagrams) over its physical connection to the local network and the Internet.

In the Internet layer, there is Internet Protocol (IP), which defines how bits and bytes of data are organized into

large groups (called as packets), and the addressing schema by which different machines find each other.

In the transport layer, there is Transmission Protocol (TCP), which is responsible for ensuring that packets are received in the order they were sent and making sure that no data is lost or corrupted. If a packet is lost, the transport layer can ask the sender to retransmit that packet.

The layer that delivers data to the user is called as the application layer. The three lower layers (the transport layer, the internet layer, and the host to network layer) work together to define how data is transferred from one computer to another. The application layer decides what to do with that data after it's transferred.

TDWS network is shown as Figure 2.8. The information

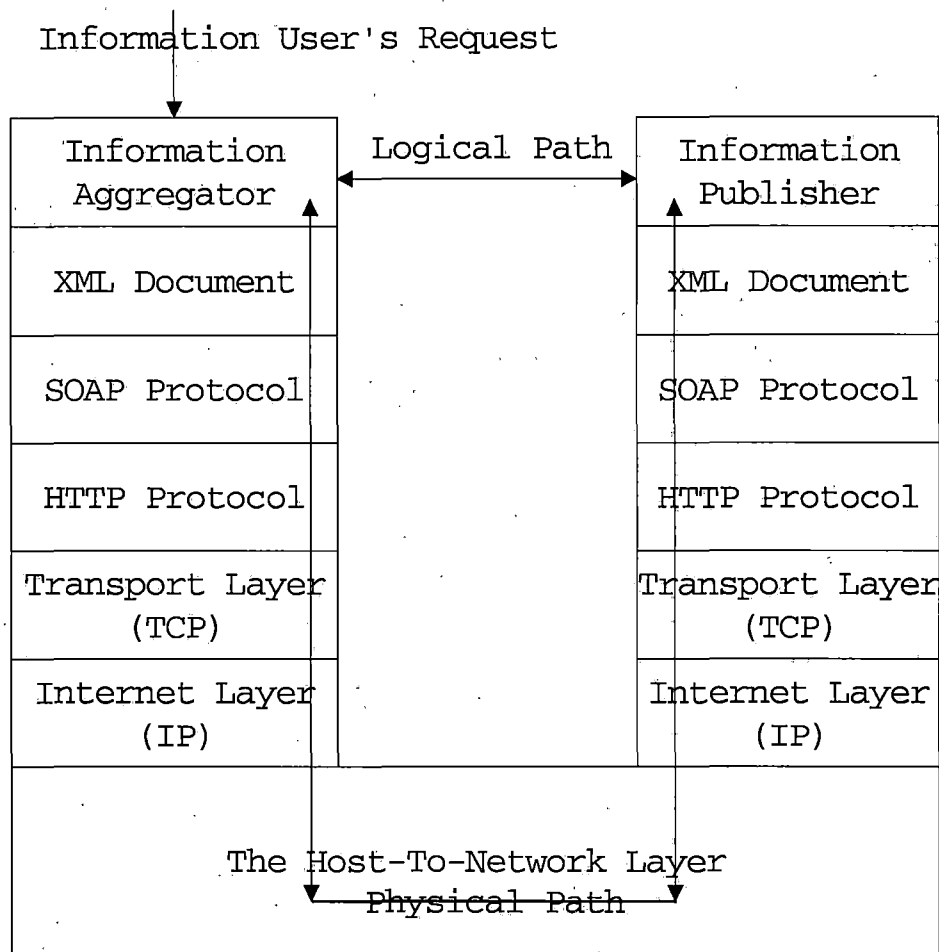


Figure 2.8. The Project Network

user goes to a Web browser and sends a Web request message for telephone numbers to the information aggregator. When the information aggregator receives that Web request from the Web browser, it creates a XML document that is to carry on the request data over SOAP protocol over HTTP protocol as a SOAP request message. Since HTTP uses TCP/IP protocol for data transfer, the information aggregator uses socket APIs to make a TCP connection to the information publisher.

After the TCP connection is established, the information aggregator sends the SOAP request message to the transport layer as a socket output stream. According to the definition of HTTP request header field and SOAP header field, the transport layer processes the SOAP request message, including breaking up the request into TCP segments, and then the transport layer passes the SOAP request message to the local Internet layer. The Internet layer fragments the segments into IP datagrams. According to the specified IP address 139.182.137.21 and the port number 9968 for Telephone Company One and 8080 for Telephone Company Two, the host-to-network layer encodes the digital data as analog signals by using iso-8895-1 and sends them to the host-to-network layer on the information publisher's computing system to which it's addressed. The host-to-network layer on the information publisher computing system decodes the analog signals into digital data and then passes them to the information publisher's Internet layer. The Internet layer then passes them to the transport layer. The transport layer checks them to see that all the SOAP request data have arrived and it requests retransmission of any missing or corrupt pieces. Once the datagrams composing all or part of the SOAP request message have been received, the transport layer reassembles the datagrams into a stream and passes the stream up to the Web service running on the information publisher computing system. The information

publisher receives the SOAP request message and uses a XML parser to parse XML document stored in a SOAP envelope and extract the actual request data. The information publisher accesses its DBMS and gets response. And then, the information publisher sends the response back through the layers on the information publisher system to the Internet. Finally, the Internet delivers the response to the information aggregator.

## CHAPTER THREE

### PROJECT DESIGN

#### 3.1 Overall Description

##### 3.1.1 Project Perspective

3.1.1.1 System Interfaces. In the TDWS system, there are three system interfaces. The first interface is located between the information user and the information aggregator, and it is HTML document over HTTP protocol. The second interface is the one between the information aggregator and the information publisher and it is XML document over SOAP protocol over HTTP protocol. The last interface presents between the information publisher and its DBMS, and it is JDBC APIs (Java Database Connectivity APIs). The DBMS is developed by using postgresql. HTTP, SOAP, XML, Java, Java Script Language, Java Server Page (JSP), Socket paradigm, JAX-RPC, JAXP and ANT will be used to implement the system interface of the TDWS system on Linux operating system. In addition, the TDWS system source codes are reusable.

3.1.1.2 Hardware Interface. The existing operating system can handle TDWS's hardware interfaces, such as Linux. The TDWS system will not implement any hardware interface.

3.1.1.3 Software Interface. In the TDWS system, software interfaces are provided by using Linux operating system, Tomcat 4.1.18 for Java Server Page and deployment of Telephone Company Two's Web service, J2SE 1.4.1 and Apache



ANT 1.5.3 for compiling Java source tree, and PostgreSQL for developing database management system. In order to access TDWS, the information user needs to use a Web browser, which is Microsoft Internet Explorer. The Table 3.1 shows the summarization of software interface.

Table 3.1. Software Interface

Software	Response
Operating System	Red Hat 7.3 Linux
Browser	Microsoft Internet Explorer
JSP server	Jakarta-Tomcat-4.1.18
JAX-RPC web server container	Jakarta-Tomcat-4.1.18
Java Compiler	J2SE 1.4.1
Java-based build tool	Apache Ant 1.5.3
Database Management System	Postgresql
Java Web Services Develop Pack (Separately installed)	Java Web Services Develop Pack 1.1
Script Language	JavaScript

3.1.1.4 Assumptions and Dependencies. In this project, there are following assumptions:

- It is assumed that there are two cities named City1 and City2 in California.

In each of City1 and City2, there are ten streets. Every two-street has the same zip code. Therefore, every

city has five zip codes. In addition, area code of telephone numbers in City1 is 908 and that in City2 is 789. The middle three-digit number could be any number except "000" and the last four-digit number could be any number except "0000".

- It is assumed that City1 uses Telephone Company One as its Telephone Company, and City2 uses Telephone Company Two as its telephone company. The Table 3.2 shows the information summarization of City1 and City2.

Table 3.2. City1 and City2

	City1		City2	
Telephone Company	Telephone Company One		Telephone Company Two	
Area Code	908		789	
Street Name With Zip Code	First Street	92111	Lake Street	92123
	14 <sup>th</sup> Street		Park Street	
	Big Road	92222	Left Road	92345
	Small Road		Right Road	
	Little Drive	92333	Moon Drive	92567
	Parkway Drive		Rain Drive	
	North Boulevard	92444	Snow Boulevard	92678
	South Boulevard		Iowa Boulevard	
	University Avenue	92555	Sunset Avenue	92890
	Mission Avenue		Main Avenue	

- There are no applicable hardware limitations.

#### 3.1.1.5 Perspective of Telephone Companies

Administrators. The telephone companies' administrators are in charge of operating and managing telephone companies' DBMS. They can insert new records into DBMS; delete records from DBMS, update, search and review records stored in DBMS.

3.1.1.6 Perspective of Information Users. The information user will be able to use Web browsers, enter the request information, and view information sent back from the information aggregator.

#### 3.1.2 Project Functions

In the TDWS system, the information publisher, which is two telephone companies that provide web services, is simulated. The information aggregator, which is a business entity presented as a telephone directory Web site whose purpose is to provide access to telephone directory data, is also constructed. The information aggregator will provide a function to the information user, which is a data entry form for the information user to request telephone numbers of people that the information user specifies. The detailed operation sequence of TDWS is described as follows: the information user goes to a Web browser and sends a Web request for telephone numbers to the information aggregator. When the information aggregator receives the Web request, it translates the Web request message into a SOAP request

message and sends the SOAP request to the information publisher. The information publisher receives the SOAP request, processes it, generates a SOAP response message, and sends the SOAP response to the information aggregator. After obtaining the SOAP response message from the information publisher, the information aggregator processes the SOAP response message, formats a HTML response message, sends the HTML to the information user, and presents the HTML to the information user through his or her Web browser.

### 3.2 Project Architecture Design

According to Scott Ambler [8] and Jeff Offutt [10], Web site design is commonly viewed as having three-tiered client-server architecture. The 3-tiered client-server architecture is shown in Figure 3.1.

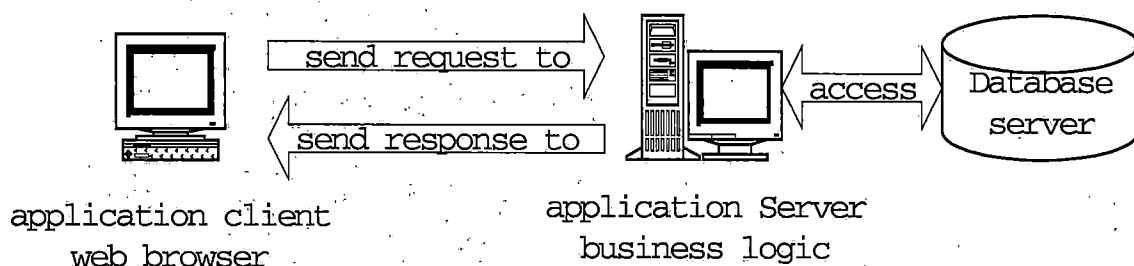


Figure 3.1. A Three-tiered Client/Server Architecture

A user component interacts with a component housing the central business logic. The central business logic in turn interacts with a data store. This is a regular model for a 3-tier architecture. The design of this project might have

some similarities to the above regular 3-tier architecture. However, an additional Web service tier (the information publisher) is created in this project to separate the data store from the business logic (the information aggregator). The purpose of this kind design is to demonstrate how the information aggregator access public information through the information publisher. In this project, the business logic passes information requests to the Web service, and then the Web service extracts information from the data store using its own logic and sends response back to the business logic.

In the TDWS system, there is two separated but cooperating systems: the information aggregator and the information publisher. Two different prototypes were developed for the information publisher: Telephone Company One (telco one) with its Web service developed manually by using socket paradigm with XML Schema, SOAP protocol, and HTTP protocol; and Telephone Company Two (telco two) with its Web service developed by using JAX-RPC technology with Java frame work. Each telephone company has its own DBMS. TDWS has three Web sites, which are developed for different users and they can be accessed with different URLs through the Internet. The information aggregator is designed to send/receive SOAP messages for telephone numbers, and the information aggregator is going to be used by the information user. The Web site of Telco one is developed only for Telco one administrators to manage Telco one's

DBMS. The Web site of Telco two is similar to the Web site of Telco two, and it is implemented only for Telco two administrators to manage Telco two's DBMS. The detailed explanation of the TDWS architecture is given in the next subsections.

### 3.2.1 The Architecture

The TDWS architecture is shown in Figure 3.2. Three components are presented: the information user, the information aggregator, and the information publisher with DBMS.

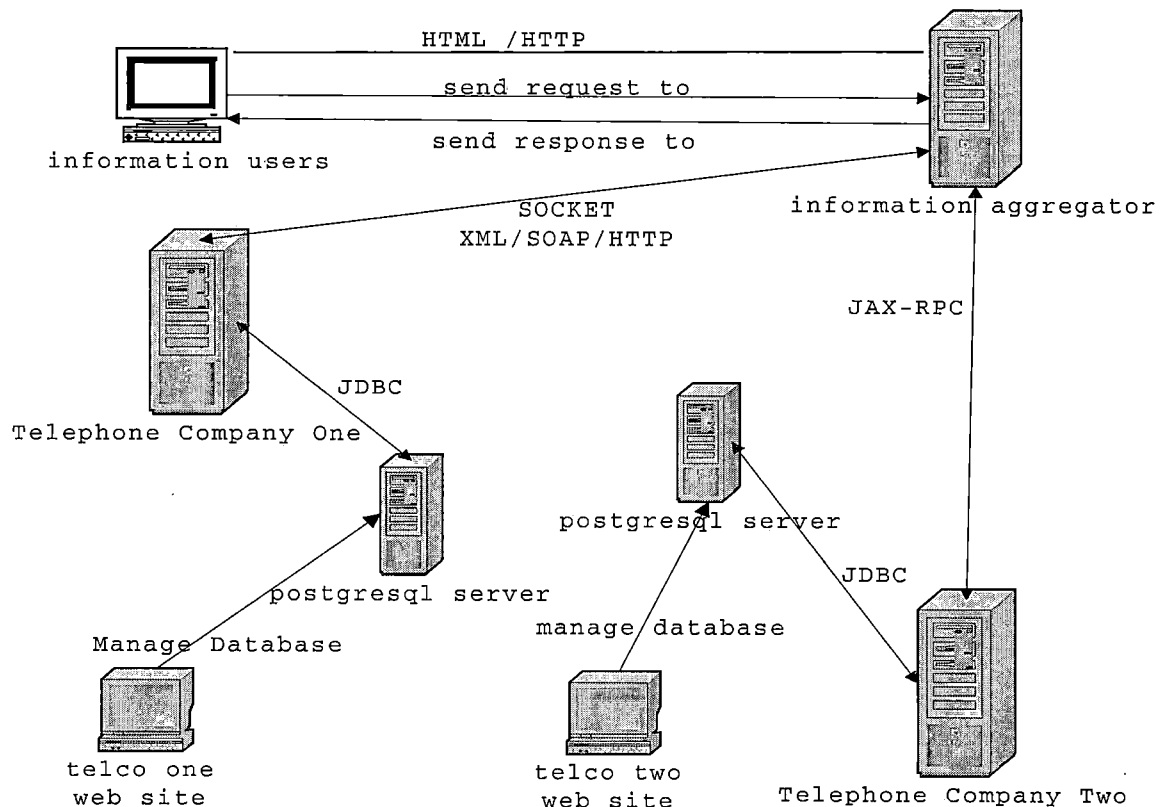


Figure 3.2. Telephone Directory Web Service

### 3.2.2 Description

The components in the Figure 3.2 are described as follows:

The information user is a person who uses a Web browser to get information from the information aggregator.

The information aggregator is a business entity. It develops an application by using Web services provided by the information publisher. The application is a telephone directory web site that provides users with a single and simple interface to all public telephone numbers. The information aggregator communicates with the information publisher through the Web Services and the application. The information aggregator receives a Web request from the information user, sends a SOAP request message to the information publisher, receives a SOAP response message from the information publisher, and finally sends a HTML response back to the information user. Examples would include general service portals, such as Google, etc. Google develops its APIs and Web service. Many other companies can develop their applications by using Google Web service to search information from Google DBMS directly.

The information publisher is Telephone Companies with their Web services. When the information publisher receives SOAP request messages from the information aggregator, it starts to access its DBMS via JDBC and looks for the requested information. Once the information publisher finds

out the results on the requested information, it will send a SOAP response message back to the information aggregator. The information publisher has its DBMS developed by using postgresql. The DBMS stores public information. Moreover, the information publisher has a Web site. Administrators can manage DBMS through Web site.

The TDWS system's life cycle is described as follows:

- The information user goes to a Web browser, fills out a HTML form with request information, and clicks "search" button. The request message will be sent to the information aggregator as a HTML document over HTTP protocol request from the web browser.

- The information aggregator takes and analyzes the Web request message. Then it generates a XML document in a SOAP envelope over HTTP protocol as a SOAP request message. Finally, the information aggregator sends the SOAP request message to the information publisher.

- Once the information publisher receives the SOAP request message, it will take following actions: a) parses the XML document with an appropriate XML parser; b) gets actual request data; c) accesses the postgresql DBMS through JDBC APIs with the request data; d) generates a XML document with response information in a SOAP envelope over HTTP protocol as a SOAP response message.



- The information publisher sends the SOAP response message back to the information aggregator via its Web service.

- The information aggregator receives the SOAP response message, parses the XML document with XML parser, obtains actual response information, formats the response message as a HTML document, sends it to the information user, and displays the HTML on the information user's web browser.

### 3.3 Information Aggregator Design

The information aggregator receives Web requests from the information user; converts Web requests into SOAP request messages, and then sends them to the information publisher. The information aggregator also receives SOAP response messages from the information publisher, converts them into HTML response messages, and sends them back to the information user.

#### 3.3.1 Use Case Diagram

Figure 3.3 shows Use Case Diagrams for the information aggregator. The Information user goes to a web browser and search phone numbers.

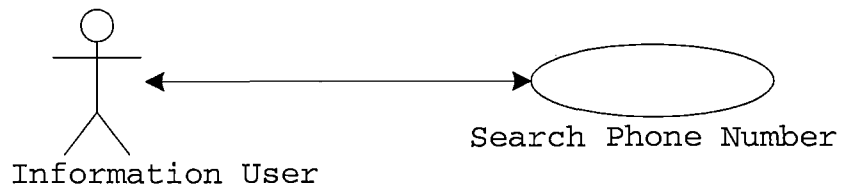


Figure 3.3. Use Case Diagram

### 3.3.2 Graphical User Interface

When the information user goes to the Web browser, the first Web page that users see is the enter page. This page is a starting page. Once users click the link "Telephone Directory Web Service", users will be able to get into the TDWS home page. TDWS has the same banner on each Web page, except the enter page of TDWS. The banner consists of the characteristic of TDWS system, and a main menu bar. The main menu bar includes "TDWS Home", "Search", and "About TDWS". The table 3.3 summarizes responses of each link and each icon to user's actions.

Table 3.3. User Actions and Responses

User Actions	Responses
Click the Telephone Directory Web Service link	Display the TDWS home page
Click the TDWS Home link	Display the TDWS home page
Click the Search link	Display the Search page with a HTML form

Table 3.4. User Actions and Responses (Continue)

User Actions	Responses
Click the About TDWS link	Display the introduction page
Click the Map link	Display a map for cities and telephone companies
Click the Javadoc link	Display a Javadoc page.
Click the codeviewer link	Display source code directory page.
Click the Search icon	Display the response information
Click the Clear icon	Clear up the request form on the search page

3.3.2.1 The Search Page. The Search page is shown in Figure 3.4. Users go to this page to search all the related information to the phone number on a specified person, including person's name, address, etc. Before users click the Search icon, users need to specify person's name and area in which the person lives. Then, users either can click Search icon to submit the request information or click Clear icon to reset input. If users specify the area field with City1, the TDWS system will go to Telco one to look for specified person's information through Telco one's Web service. If users specify the area field with City2, the

TDWS system will go to Telco two to look for information through Telco two's Web service. If information users don't know the area in which the requested person lives, users can leave the area field with "area". And then TDWS system will go to both telephone Companies to search information through telephone companies Web service. If users don't specify the first name and middle name (the last name entry is required), TDWS will show all persons' information associated with the last name. All names could be partial names.

The screenshot shows a web browser window titled "TDWS-Telephone Directory Web Service-Master Project CSUSB-Microsoft Internet Explorer". The address bar shows "http://tdws.las.csusb.edu:8080/search/index.jsp". The page content includes the TDWS logo and title, a navigation bar with "TDWS Home", "Search", and "About TDWS", and a search form titled "Search Telephone Number" with the note "(\*All fields are required)". The form has a section "Enter Information" with input fields for "First Name:", "Middle Name:", "Last Name\*:" (containing "s"), and "Area:" (containing "area"). There are "Search" and "Clear" buttons at the bottom of the form. The Windows taskbar at the bottom shows the Start button, several open applications including "zuoYibody1 - Micro...", and the system clock showing "1:01 AM".

TDWS-Telephone Directory Web Service-Master Project CSUSB-Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites History

Address http://tdws.las.csusb.edu:8080/search/index.jsp Go

**TDWS**  
Telephone Directory Web Service  
http://tdws.las.csusb.edu:8080/tdws

TDWS Home Search About TDWS

Search Telephone Number  
(\*All fields are required)

Enter Information

First Name:

Middle Name:

Last Name\*:

Area:

Search Clear

Start zuoYibody1 - Micro... TDWS-Telephon... Internet 1:01 AM

Figure 3.4. The Search Page

3.3.2.2 The Javadoc Page. The Javadoc page is shown in Figure 3.5. Javadoc is the tool, from Sun Microsystems, used to generate API documentation in HTML format from doc comments in source code. This project generates java source code documentation using the javadoc tool and ANT.

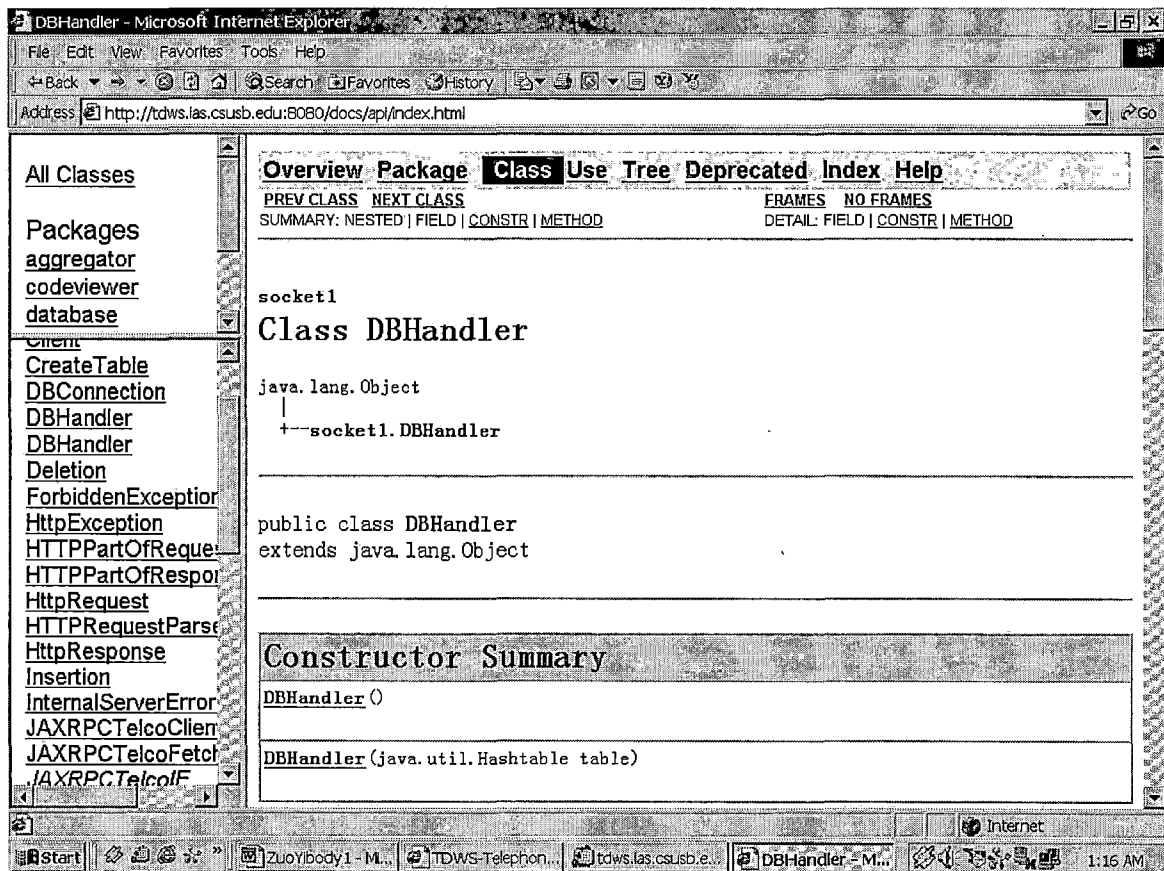


Figure 3.5. The Javadoc Page

3.3.2.3 The Codeviewer Page. The codeviewer page is shown in Figure 3.6. The page overviews all java source codes associated with this project. To view the source codes, users can simply click "codeviewer" link.

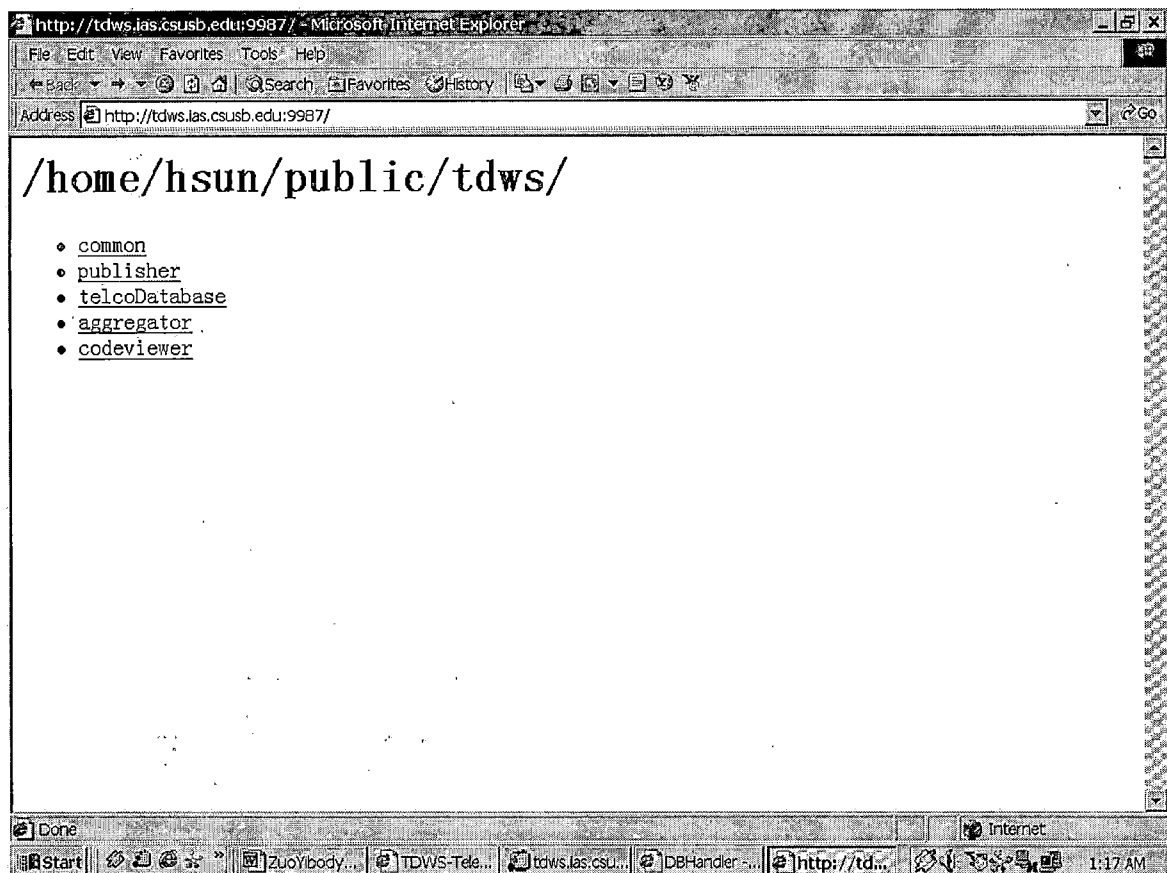


Figure 3.6. The Codeviewer Page

### 3.4 Information Publisher Design

In this project, two telephone companies are presented as the information publishers: one is Telco one with its Web service, which is implemented manually by using Socket paradigm and JAXP technology; and the other one is Telco two with its Web service, which is implemented by Web service technology JAX-RPC with Java frame work. Each telephone company is co-designed with its DBMS and its private Web site. Detailed design is provided in following sections.

### 3.4.1 Telephone Company One

Telco one is developed by using Socket paradigm manually with the Client/Server model shown in Figure 2.5. Telco one receives SOAP request messages, accesses its DBMS, and generates SOAP response messages. Finally, Telco one sends the SOAP response messages to the information aggregator. Telco one administrators can manage DBMS via Telco one's Web site. The diagram of Telco one is shown in Figure 3.7.

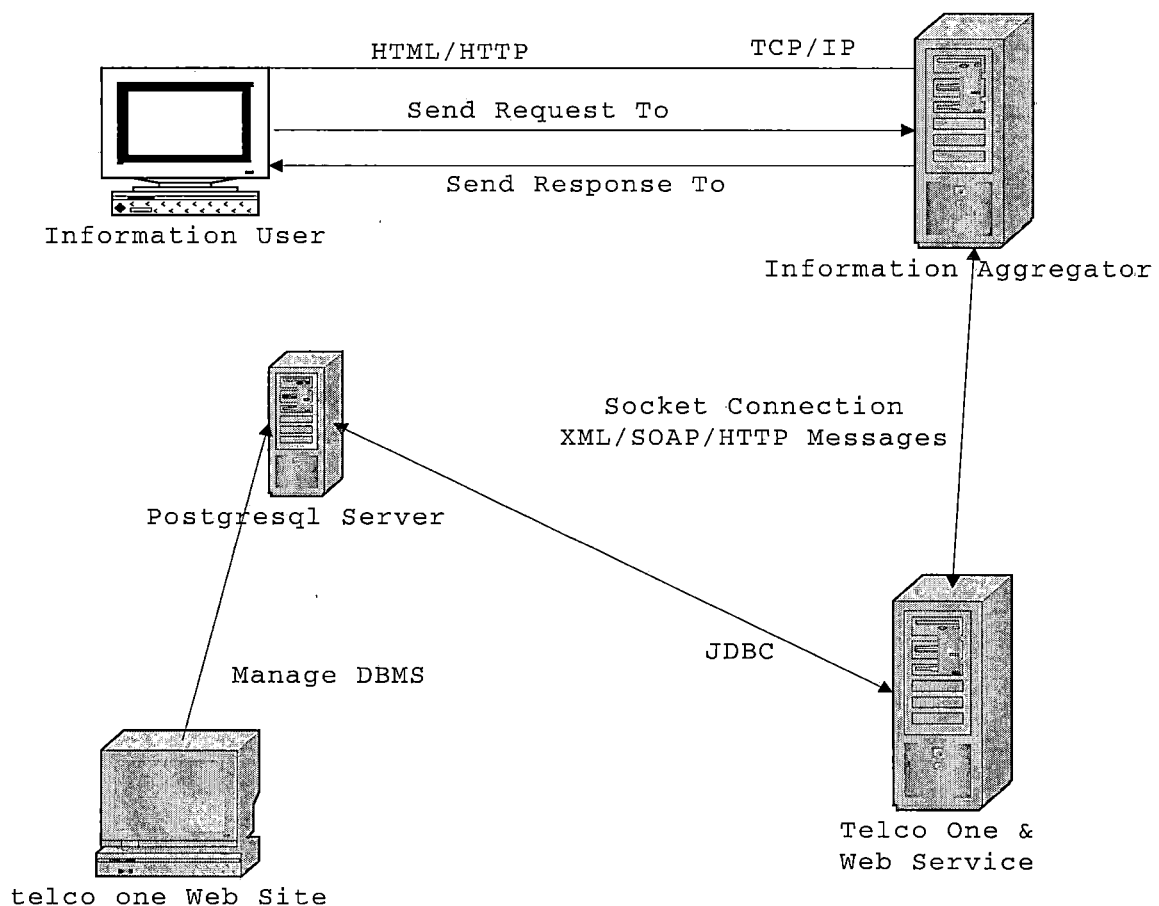


Figure 3.7. Telephone Company One's Web Service

3.4.1.1 Database Management System Design. Telco one is designed with its DBMS. DBMS is used primarily for Telco one public data storage and retrieval. Postgresql is used to develop DBMS. DBMS will not be destroyed under any situation after they are created. The database structures also will not be changed under any situation after DBMS is created. The related database schema diagram and ER diagram are shown in Figure 3.8 and Figure 3.9, respectively.

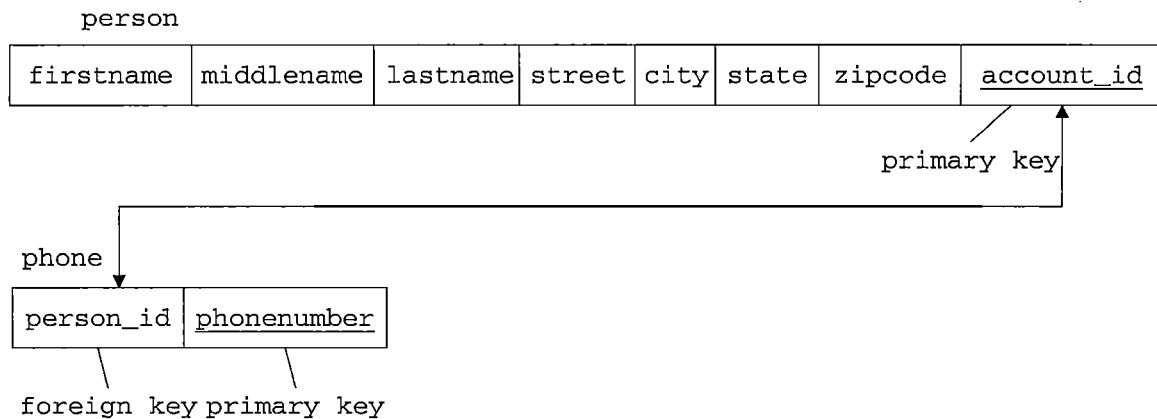


Figure 3.8. Relational Database Schema Diagram

In DBMS, there are two tables—person and phone, which store customers' information and customers' telephone numbers of Telco one.

- Person table is used to store consumers' name, address, and account\_id. Each consumer's name consists of firstname, middlename, and lastname. Each consumer's address consists of street name, city, state, and zipcode. The account\_id in person table is consumers' id and it is



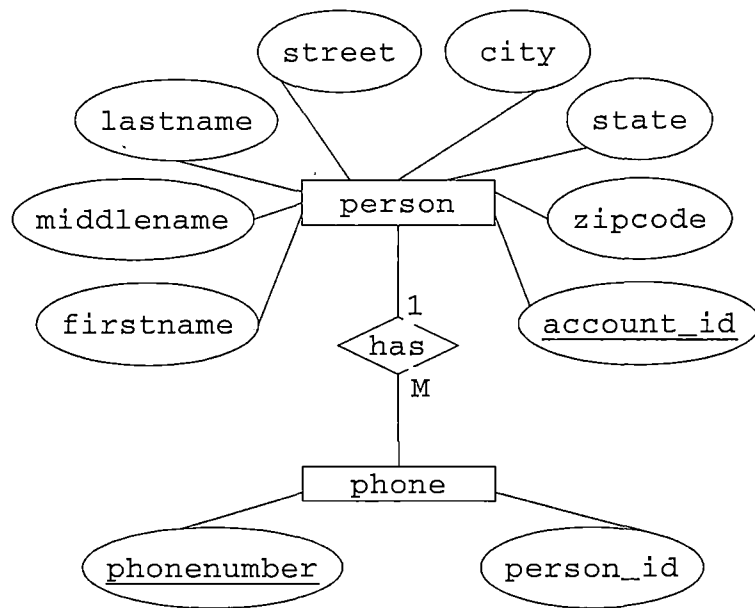


Figure 3.9. Entity Relationships Diagram

primary key of person table. The `account_id` is datalessly created. The `account_id` will be automatically incremental by one when a new record is added. For example, if the person table is empty and when a new record is added, the `account_id` will be one. If the 100<sup>th</sup> record is added, the `account_id` will be 100. It is a reasonable and recommended way to create account id. And it is the way different from that used to create account ID in real world. Many companies in the real world created account ID associated with information of customers. Someone else can get to know the customers information from his or her `account_id`. The requirement for customers' name, address and telephone number is shown in Table 3.5.

Table 3.5. The Requirement of Customers' Information

Information Name	Requirement
First Name	Regular name in English
Middle Name	Regular name in English
Last Name	Regular name in English
Street Number	Digital Number
Apartment Number	A English Letter or Digital Number
Telephone Number	Digital Number

- Telco one uses phone table to store phone number and person\_id. Phone number is a primary key of phone table and person\_id is a foreign key as a reference to person table.

- Each person can have one or more phones. That means each customer can have one or more than one phone numbers.

3.4.1.2 Web Site Design. Telco one has its Web site, and only Telco one's administrators can access Telco one's Web site. Administrators can manage Telco one's DBMS via the Web site. The Web site is designed with insertion, deletion, update, search, and review operations. The Web site has a login page, which is used to verify if users have permission to access the Web site.

Insertion operation is used to insert a new customer into DBMS. Web process of insertion function is shown below:

- A Telco one administrator enters a new customer's information, including name, phone number, address.

- Telco one system will check the input data. If the input data are invalid, the system will ask the administrator to correct the invalid information until all input data are valid.

- Before inserting a new record, Telco one system will check DBMS to make sure if the customer is a new one or it has already existed in DBMS. If the customer has existed already, Telco one system will check if the customer's telephone number is a new number or an existed one. If the telephone number is a new one, Telco one system will add the phone number only. Otherwise, Telco one system will display a message to show the record with the phone number that has already existed in DBMS.

The Figure 3.10 shows the process of insertion function.

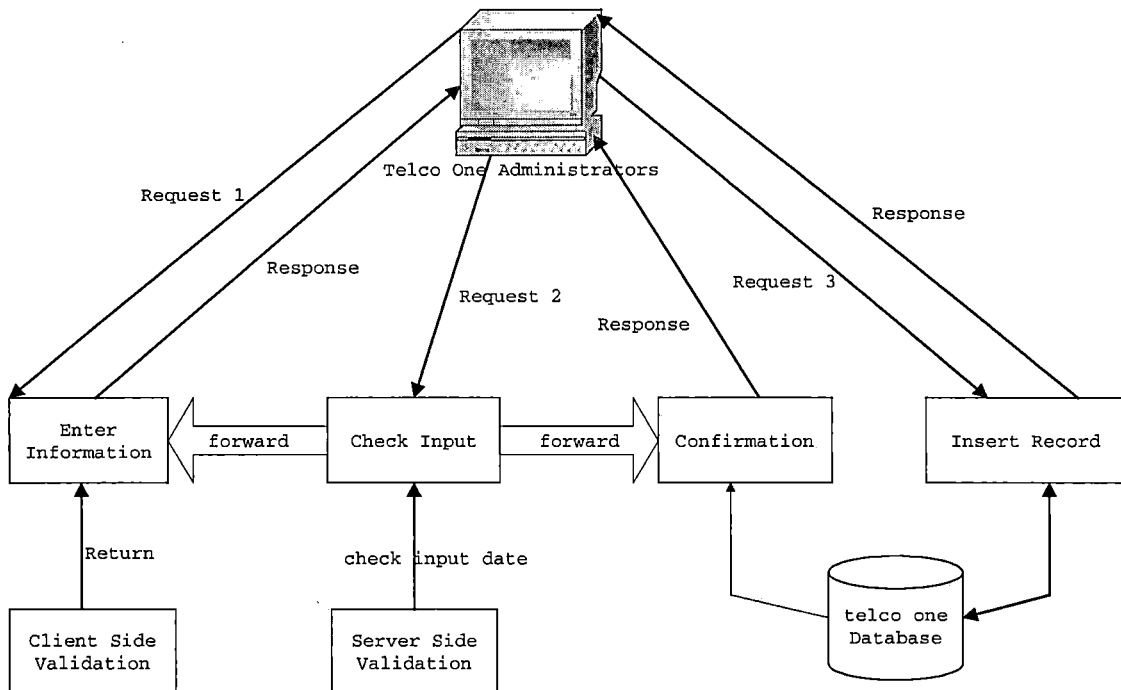


Figure 3.10. Web Process of Insertion Function

Deletion operation is to delete a record from DBMS. The web process for deletion function is shown in Figure 3.11. Deletion Web page provides a HTML form and lets Telco one administrators input a customer's first name, middle name, last name, or phone number. After administrators input data and click "Search" button, the system will go to Telco one DBMS to search if any record matches the specified names. If there are some customers existing with the specified information in DBMS, the system will display the records in a HTML table. And there is a "Delete" button on each row. Once administrators click "Delete" button, a confirmation page for the selected record is shown up with an OK and a Cancel buttons. If administrators click OK

button, the system will delete the selected record from Telco one DBMS. In this project, DBMS used to store public information does not need to connect to other databases, such as billing database. Therefore, records can be completely deleted from DBMS. It is not necessary to keep unused records in DBMS.

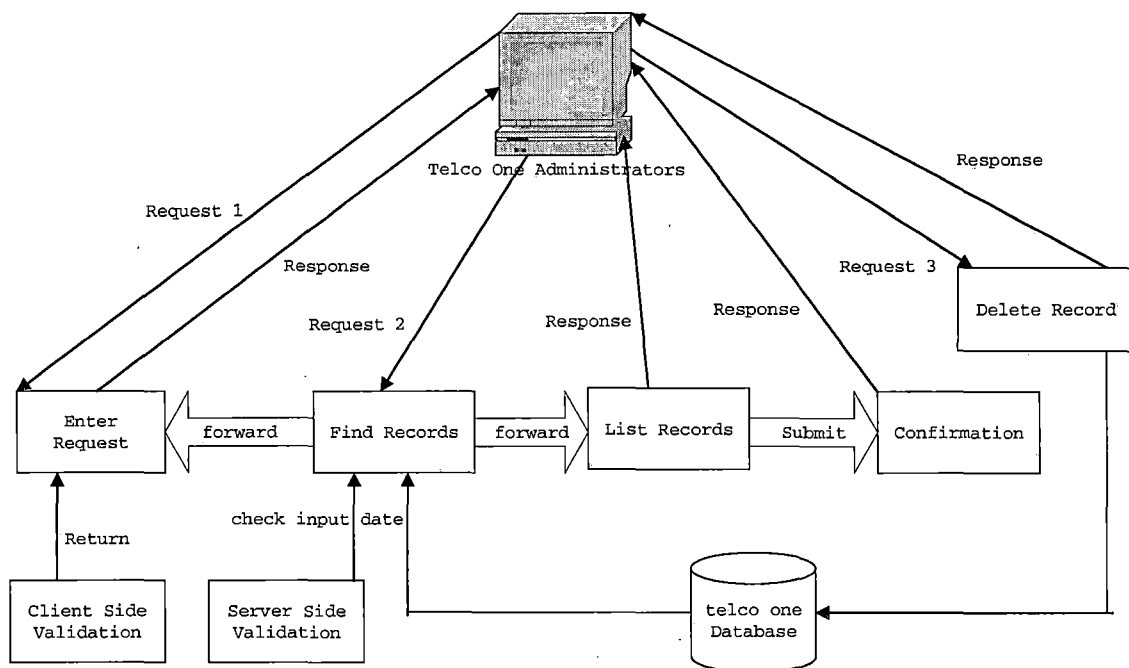


Figure 3.11. Web Process of Deletion Function

This web site also provides functions for administrators to update DBMS. Records can be edited without keeping original information on the specified records. Table 3.6 summarizes responses of each link and each icon to user actions.

Table 3.6. User Actions and Functions

User Actions	Responses
Click the Insertion link	Display the insertion menu
Click the Deletion link	Display the deletion menu
Click the Update link	Display the Update menu
Click the Review link	Display all records stored in DBMS
Click the Search link	Display a HTML Form to input data
Click the Logout link	Close current Web page and open login page
Click the Login icon	Display the main menu page
Click the insert icon	Add a new record into DBMS
Click the delete icon	Delete a specified record from DBMS
Click the Edit icon	Edit a specified record
Click the Cancel icon	Go back the menu page to choose next action
Click the Save icon	Save the new information
Click the OK icon	Confirm the information
Click the Search icon	Search DBMS with input data

The use case diagram and graphical user interface are presented as following sections.

3.4.1.2.1 Use Case Diagram. The Use case diagram for Telco one is shown in Figure 3.12.

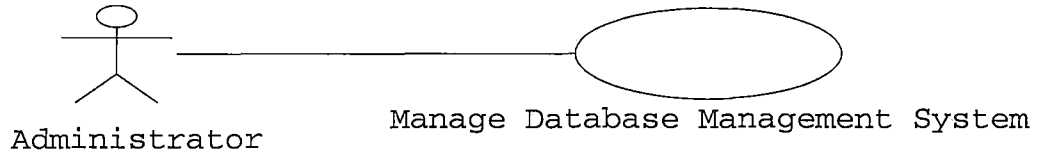


Figure 3.12. Use Case Diagram of Administrators

3.4.1.2.2 Graphical User Interface. Telco one administrators can manage DBMS through Telco one's Web site. Administrators can insert a new record into, delete a record, update a record, search records, and review records.

3.4.1.2.2.1 The Login Page. The login page is shown in Figure 3.13. This page is the starting page of Telco one Web site. Administrators need to enter correct User ID and password in order to go to main menu page.

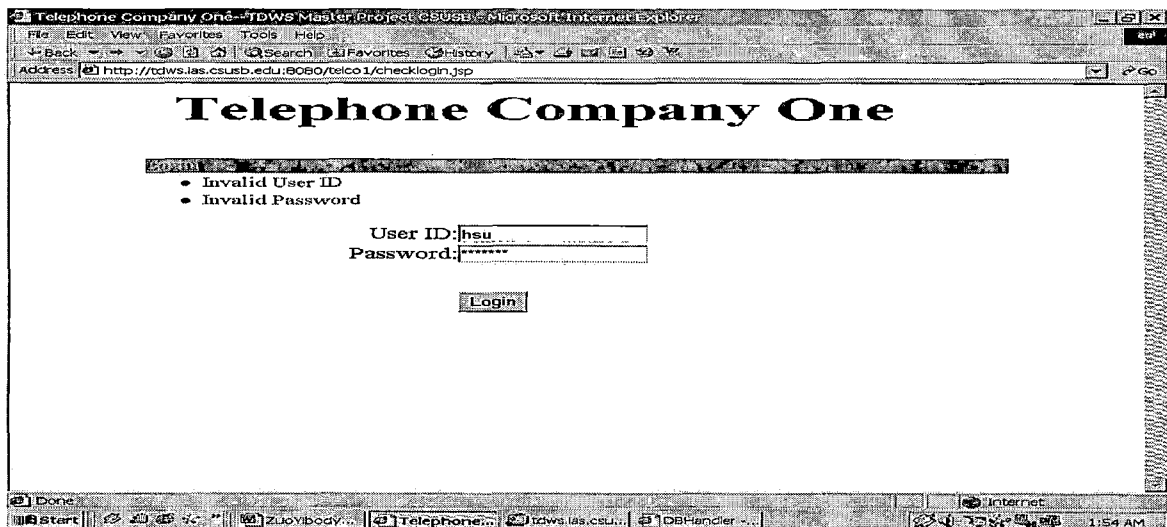


Figure 3.13. The Login Page

3.4.1.2.2.2 The Menu Page. The menu page is shown in Figure 3.14. The menu page provides users a list of menu that they can choose.

- Insertion
- Deletion
- Update
- Review
- Search

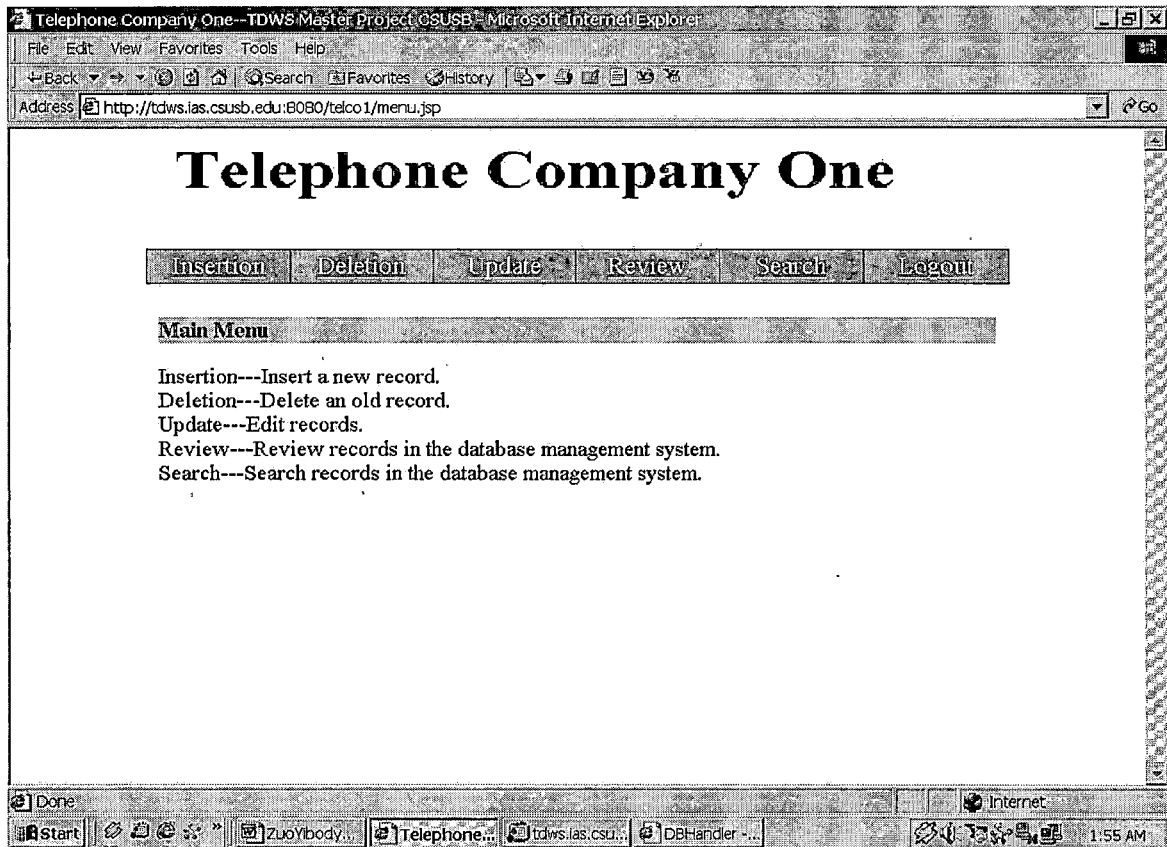


Figure 3.14. The Menu Page



3.4.1.2.2.3 The Insert A New Record Page. As shown in Figure 3.15, the "Insert A New Record Page" provides administrators a HTML form. Administrators enter information of a new record into the HTML form and then click "Insert" button to insert the new record into DBMS.

Telephone Company One--TDWS Master Project CSUSB - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History

Address http://tdws.las.csusb.edu:8080/telco1/insertion/person/validate.jsp Go

# Telephone Company One

Insertion Deletion Update Review Search Logout

## Insert A New Record

(\*All fields are required)

### A New Record

- Invalid Street Name

First Name\*: Ann

Middle Name:

Last Name\*: Kuns

Phone Number\*: 908 - 333 - 8888

Street (Number)\* 000 Select Street APT

Area\*: City1

Insert

Done Internet

Start ZuoYibody... Telephone... tdws.las.csu... DBHandler... CSCI 322 : ... 2:27 AM

Figure 3.15. Insert A New Record Page

3.4.1.2.2.4 Insert Telephone Number Page. This page is shown in Figure 3.16. The page provides a HTML form for administrators to input names. Once administrators click "Insert" button, a list page of all records associated with input data is shown up. Administrators can select customers from the list of page and insert a new telephone number for the selected customers. The list page is shown in Figure 3.17.

Telephone Company One--TDWS Master Project CSUSB - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites History

Address http://tdws.las.csusb.edu:8080/telco1/insertion/phone/find.jsp Go Links

# Telephone Company One

Insertion Deletion Update Review Search Logout

## Insert Telephone Numbers

The Record

- The specified customer could not be found

First Name:

Middle Name:

Last Name:

Phone Number: 908 -  -

Search

Start 1:td... 2:td... 3:td... 4:td... 5:td... Tele... Zu6... Internet 11:26 PM

Figure 3.16. Insert Telephone Number Page

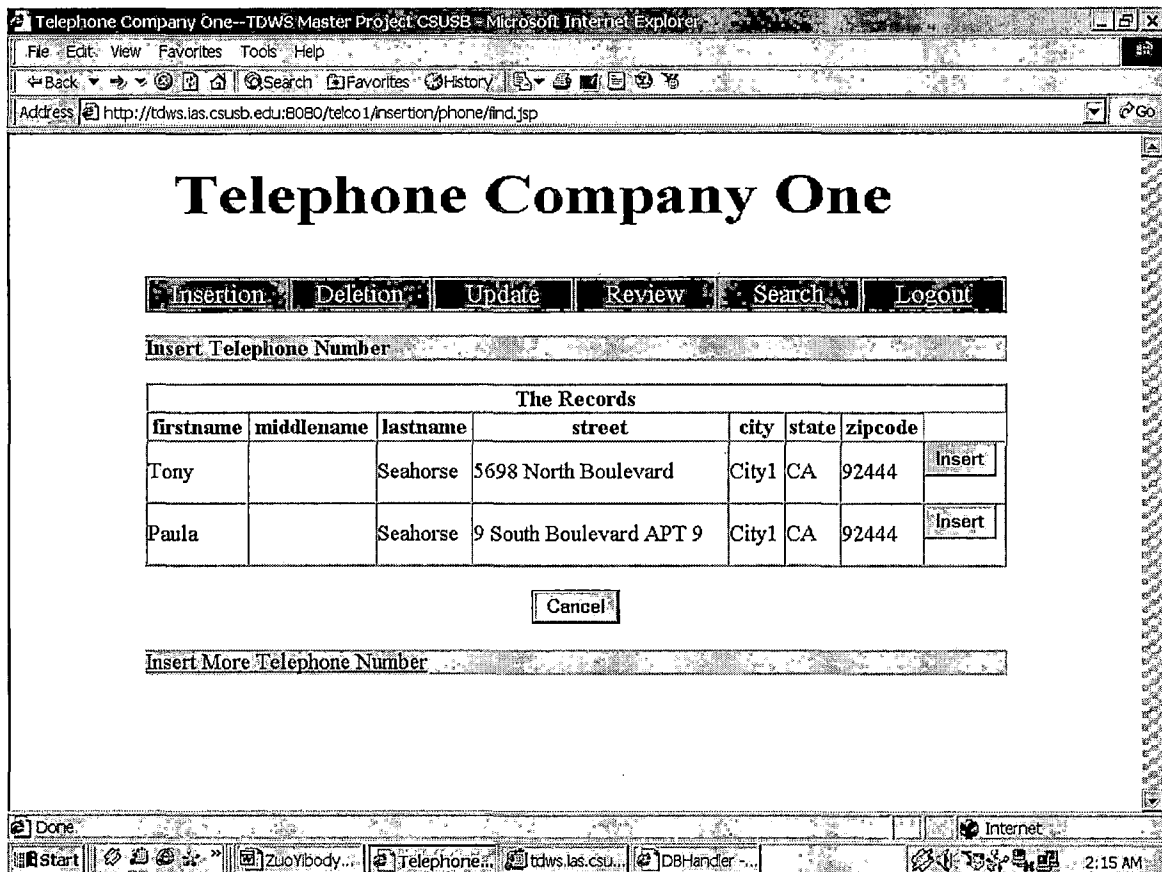


Figure 3.17. The List Page of Insert Telephone Number

3.4.1.2.2.5 The Delete Customer Page. The delete customer page is shown in Figure 3.18. Administrators use this page to delete an existed customer from DBMS. Administrators enter names of customers and click "Delete" button. A list page of records associated with the specified customers is shown up. Administrators can select a record and delete it. The list page of deleting customers is shown in Figure 3.19.

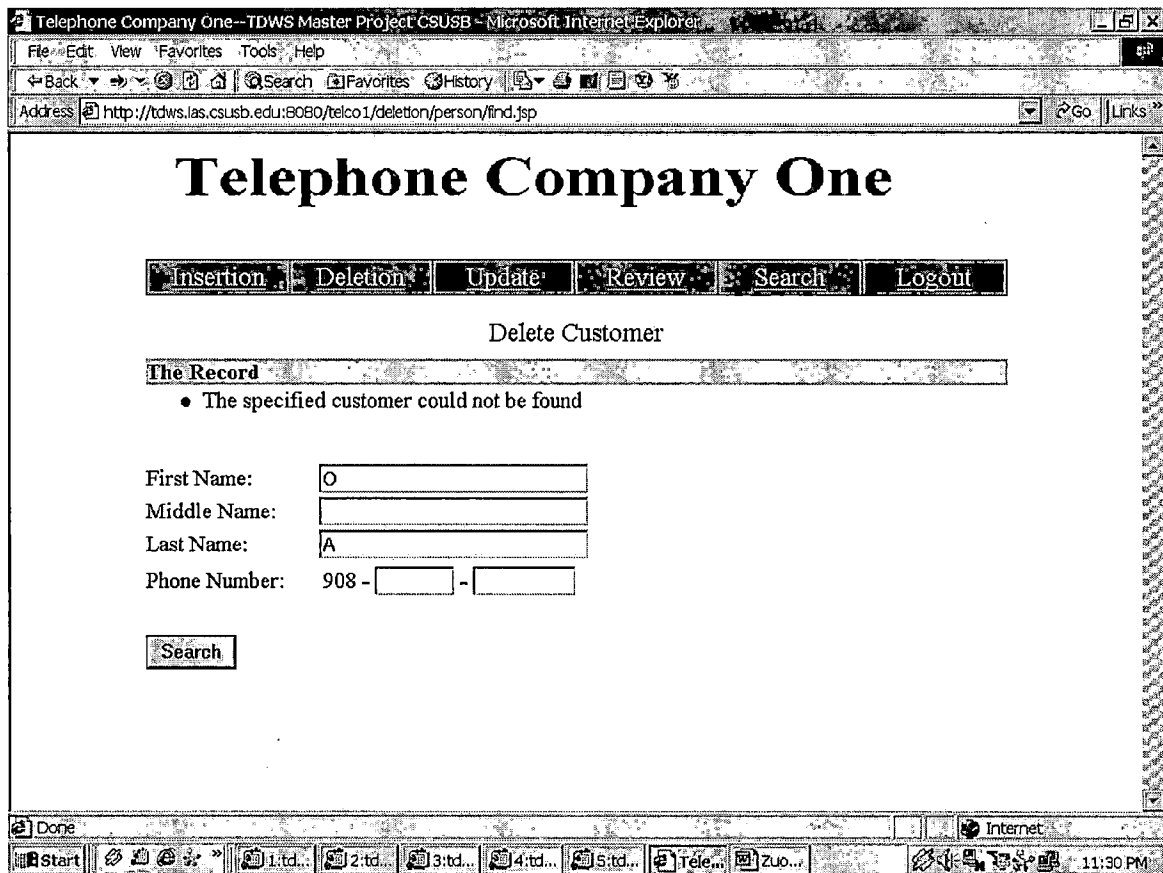


Figure 3.18. The Delete Customer Page

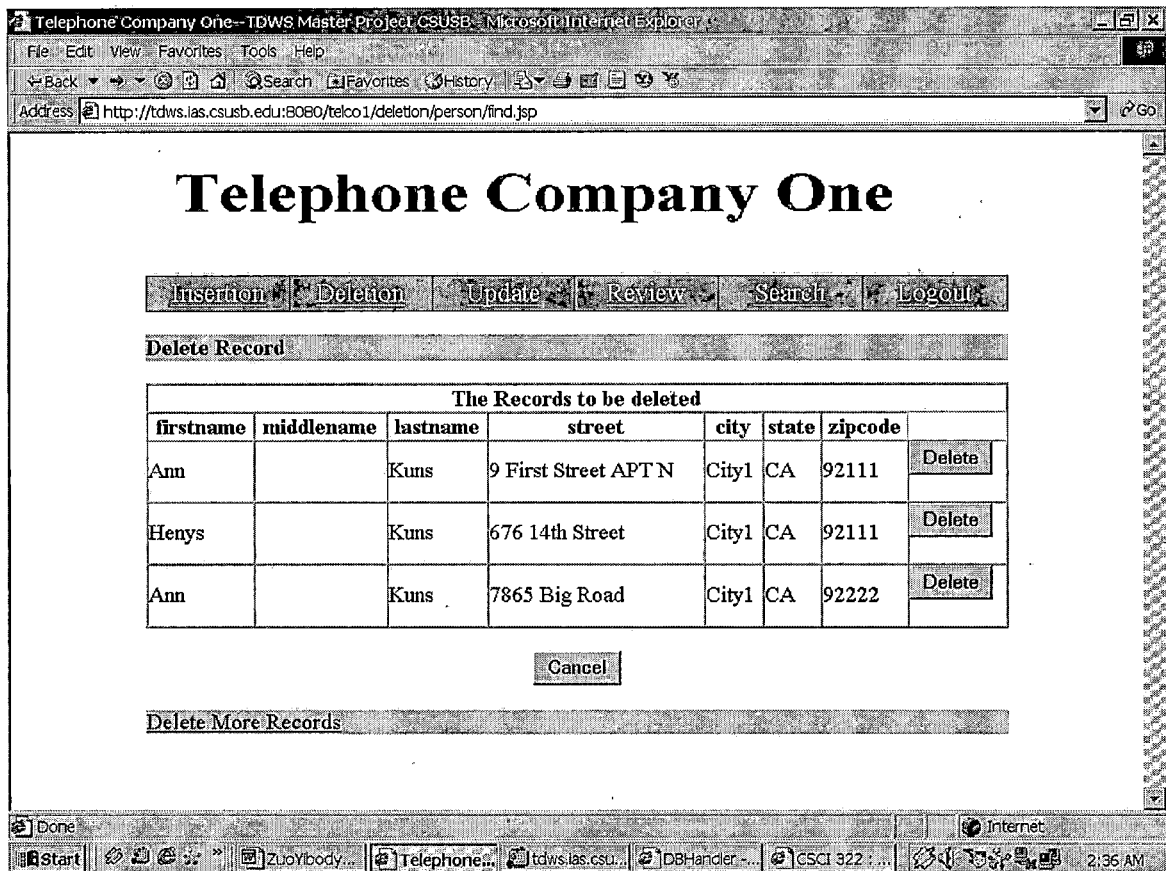


Figure 3.19. The List Page of Delete Customer

#### 3.4.1.2.2.6 The Delete Customer's Phone Number Page.

This page is shown in Figure 3.20. If a customer has more than one telephone number and the customer want to delete one of them, administrators will use this page to delete the specified customer's telephone number. If the customer has only one telephone number, administrators not only delete the customer's phone number, but also delete the customer from DBMS. The list page is shown in Figure 3.21.



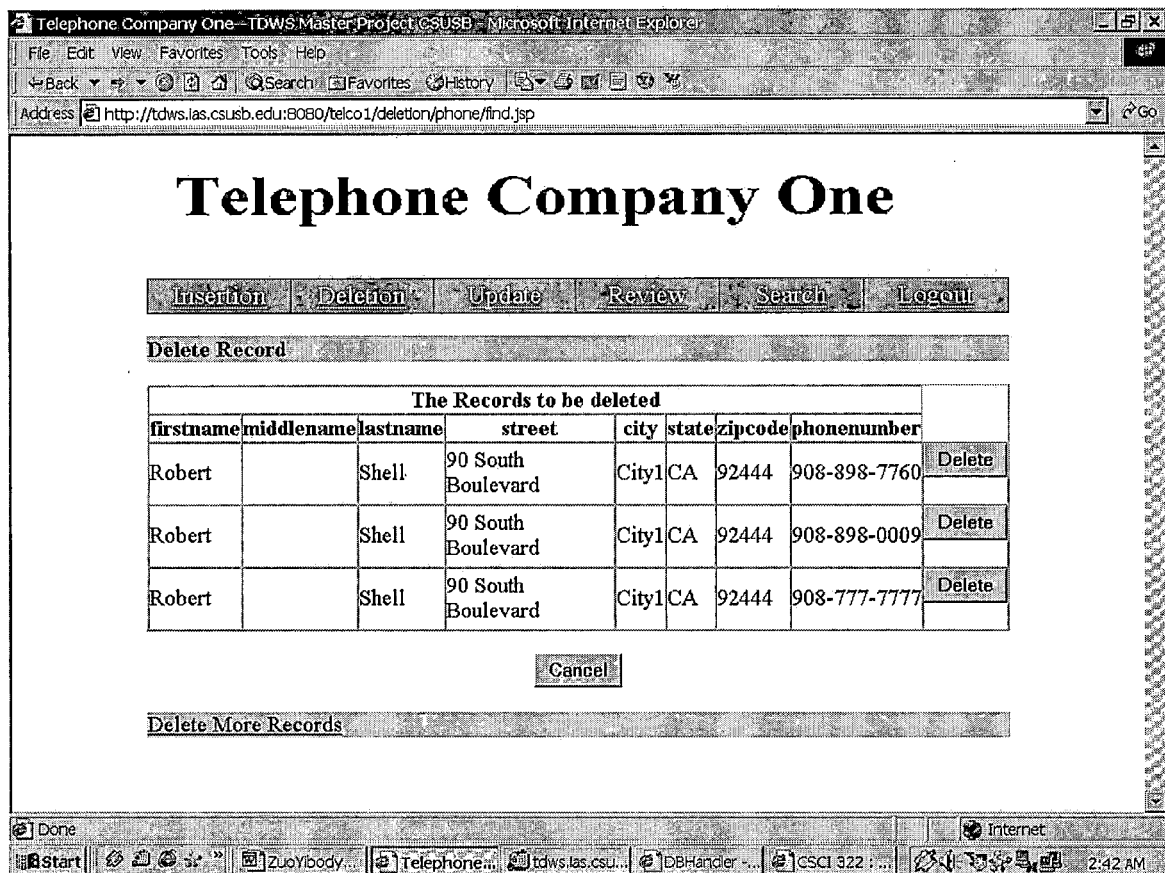


Figure 3.21. List Page for Deleting Phone Number

3.4.1.2.2.7 Edit Customer. This page is shown in Figure 3.22. This page provides a HTML form and let administrators to input customer's names. Once administrators click "Edit" button, a list page with specified customer is shown up. Administrators can operate this page and edit names, addresses or telephone numbers of the specified customers.

Telephone Company One—TDWS Master Project CSUSB—Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites History

Address http://tdws.las.csusb.edu:8080/telco1/edit/name/index.jsp Go Links

# Telephone Company One

Insertion Deletion Update Review Search Logout

Edit Name and Address

**The Record**

First Name:

Middle Name:

Last Name:

Phone Number: 908 -  -

Search

Done Internet

Start 1:td... 2:td... 3:td... 4:td... 5:td... Zuo... Tele... 11:32 PM

Figure 3.22. Edit Customer Page

3.4.1.2.2.8 The Search Page. This page is shown in Figure 3.23. Administrators input partial names or phone numbers to search customers' information.



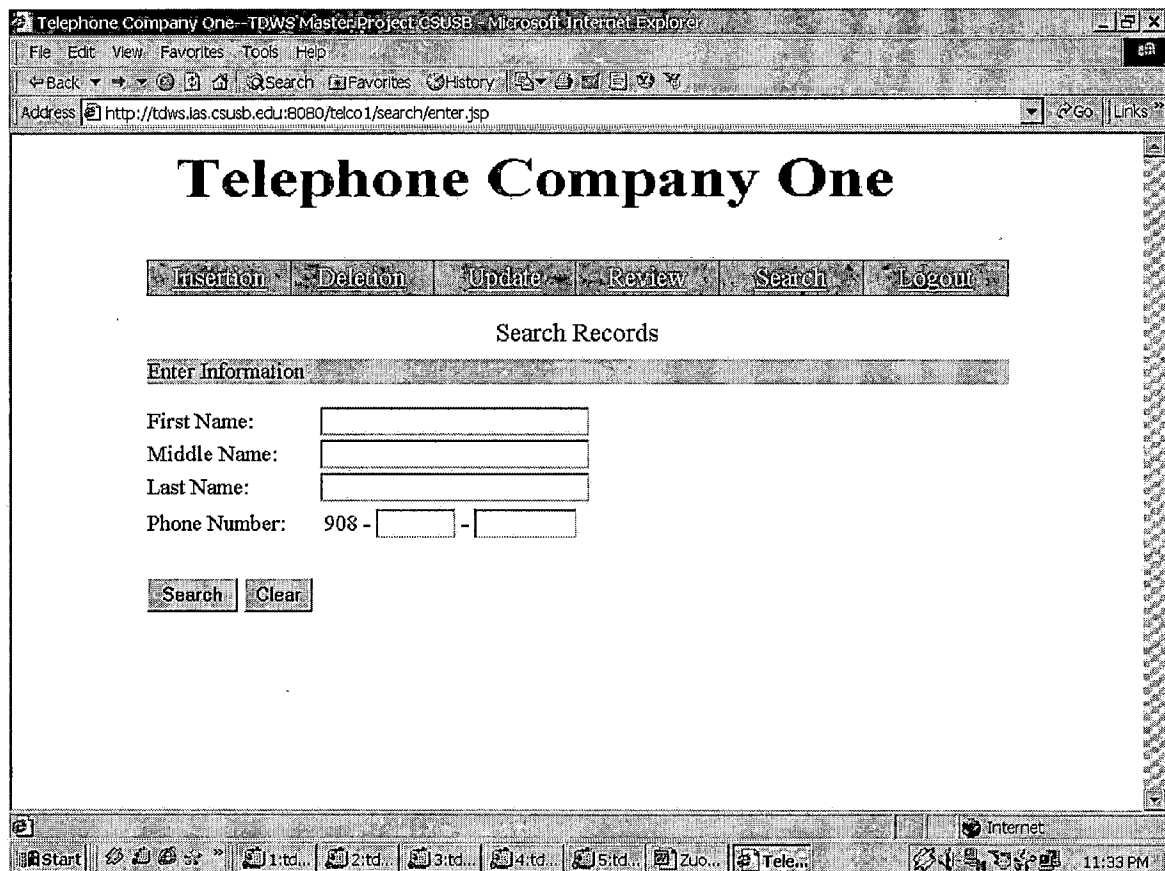


Figure 3.23. The Search Page

3.4.1.2.2.9 The Review Page. The page is shown in Figure 3.24. Administrators use this page to study records stored in each table of DBMS. This page shows users all records in each table with HTML tables.

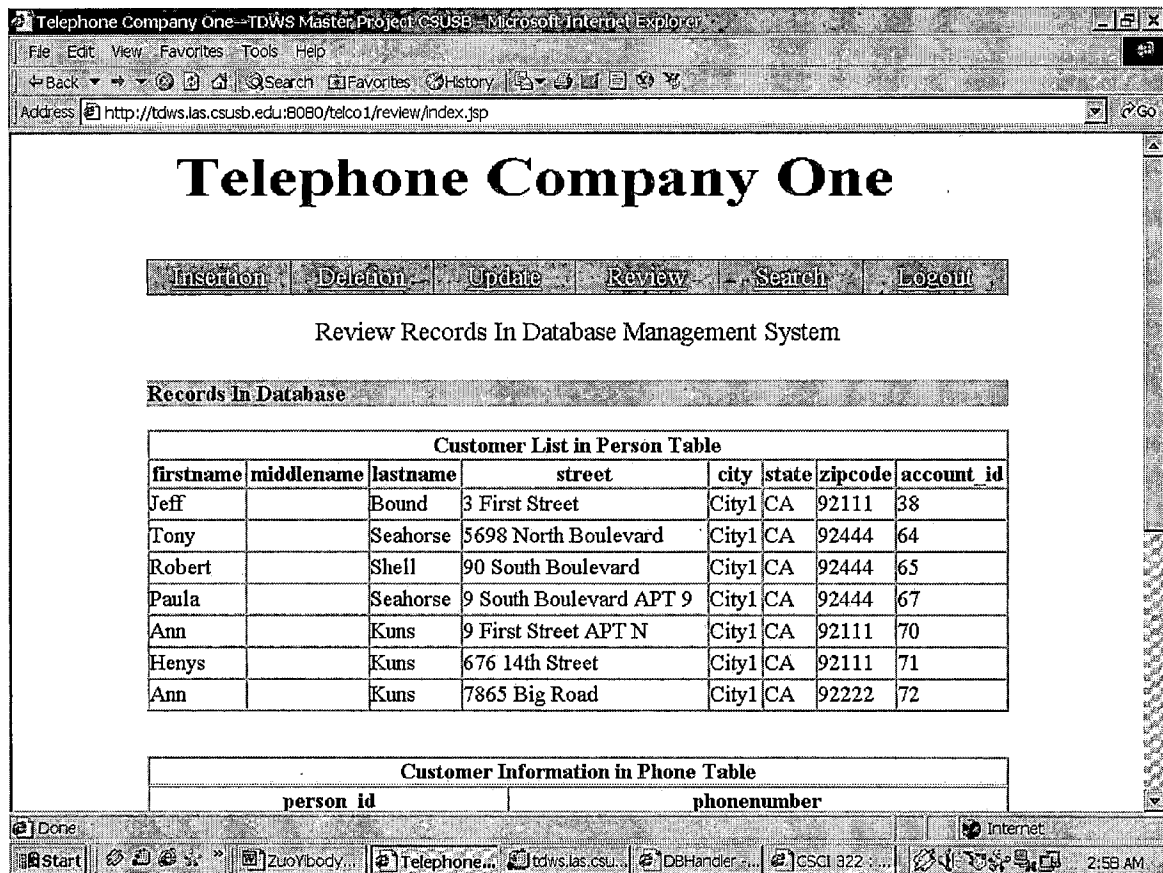


Figure 3.24. The Review Page

### 3.4.2 Telephone Company Two

In the above section, Telco one with its Web service is presented in detail. To complete the construction of the information publisher in this project, one more telephone company is needed. Hence, Telco two with its Web service is in turn designed.

3.4.2.1 The Diagram Of Telephone Company Two. Telco two with its Web service is an independent system. It has its own Web site and DBMS. A remote client of Telco two's Web service—the information aggregator—can invoke the

"getPhoneNumber" method on the remote Telco two's Web service, which accepts several string parameters and then return method response. The diagram of Telco two with its Web service is shown in Figure 3.25.

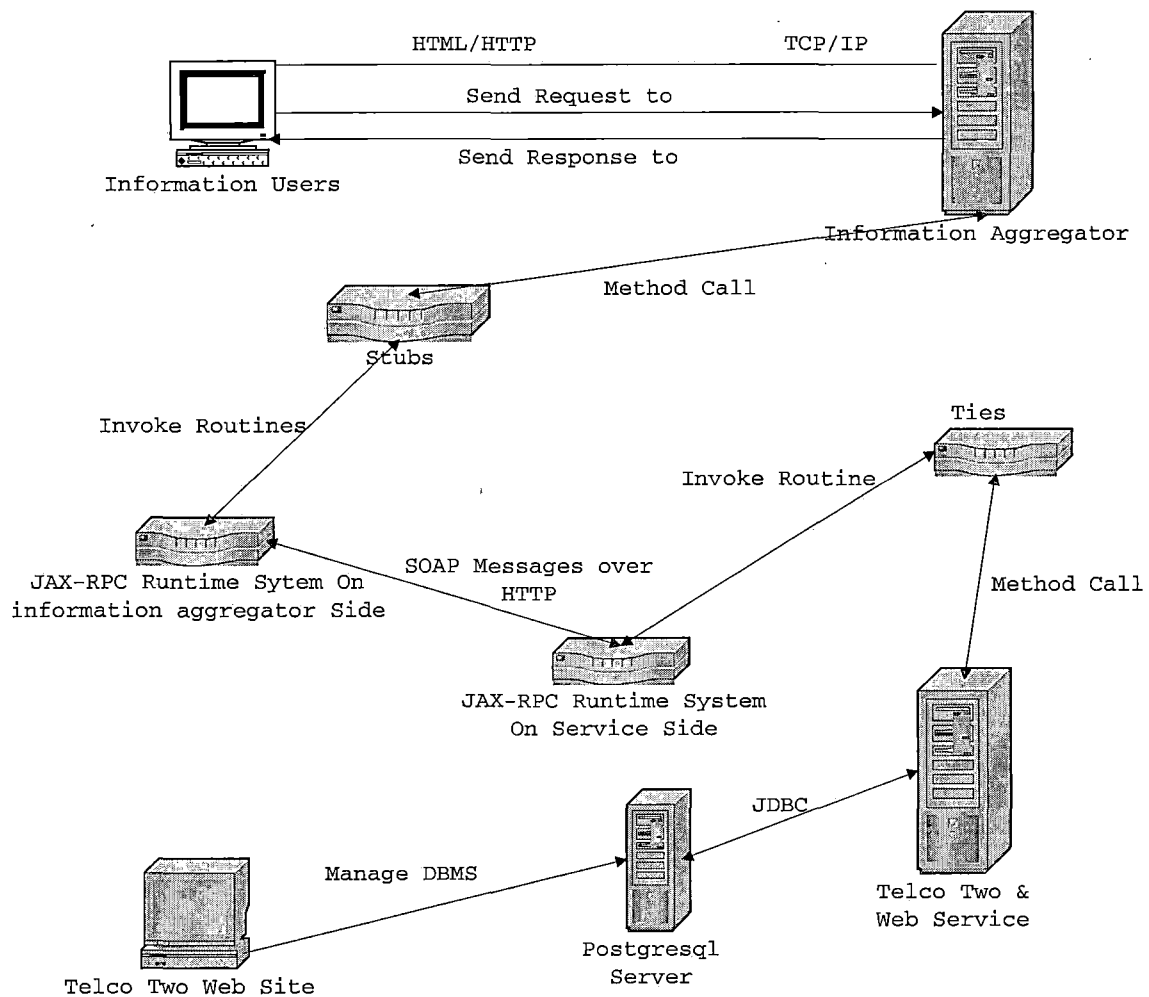


Figure 3.25. Telephone Company Two's Web Service

3.4.2.2 Database Management System Design. Telco two administrators store public information of customers into DBMS. They also retrieve information from the database.

DBMS is developed by using postgresql. The database structures will not be changed under any situation once it is created. Telco two's database management system design is similar to that of Telco one. Please go to section 3.4.1.1 and its subsections for more information on database management system design.

3.4.2.3 Web Site Design. Telco two's web site is designed for administrators to manage databases. The web site design is similar to that of Telco one. Please go to section 3.4.1.2 and its subsections for more information on the web site design.

### 3.5 Advantages

Through using TDWS, users can easily find information through the Internet. TDWS demonstrates the application of Web services on providing convenient and cost-effective access to public telephone directory data.

The advantage of the telephone directory web services model for the information publisher is that they can publish their information with lower cost. The Information publisher doesn't need to build any public Web site to publish its public information storage to the information user. The information publisher only needs to build private Web site for its administrators to manage its DBMS. Other business entities will build Web sites to access the information publisher' public information storage, such as

the information aggregator with its telephone directory Web site. This is accomplished by replacing complicated presentation logic within the information aggregator tier with simple content-only Web service APIs.

The advantage of the Web service model for the information aggregator is that the information aggregator can eliminate the need to maintain local data stores, and avoid the problem of serving stale data to the information user. That means the information aggregator does not need to create a database to store data and does not need to update the database frequently.

The advantage of the Web service model for the information user is that information is more easily obtained through the information aggregator, and information is not stale. That means the information user can get information directly from information origin.

## CHAPTER FOUR

### PROJECT IMPLEMENTATION

#### 4.1 Project Classes Design

Since last chapter describes the design of TDWS, implementing the TDWS system is in turn the next step. Unified Modeling Language (UML) is used on project analysis and design. Last chapter utilizes Use Cases Diagram. In this chapter, class diagrams will be used. As noted from last chapter section 3.2, this project has two separate subsystems: the information aggregator and the information publisher. The detailed implementation is given as follows.

##### 4.1.1 Information Aggregator

The classes of information aggregator are summarized in Table 4.1, and the class diagram is shown in Figure 4.1.

Table 4.1. Information Aggregator Classes

Classes Name	Function
SearchPhoneNumber	Super class. It receives requests from information users and sends HTML response to information users.
TelcoClient	Sending request and receive response
SocketTelcoClient	STCWS client. Sending request and receives response
JAXRPCTelcoClient	JTCWS client. Making a method call on Stubs and receives response from stubs

Table 4.2. Information Aggregator Classes (Continue)

Classes Name	Function
Request	Generating SOAP request messages.
HTTPPartOfRequest	Generating HTTP protocol header fields of SOAP request messages.
SOAPXMLPartOfRequest	Generating XML document in a SOAP envelope of SOAP request messages
Response	Separating SOAP response messages into HTTP protocol part, XML document and SOAP envelope part and generate HTML response messages.
SOAPXMLResponseParser	Using XML parser to parser XML/SOAP part of SOAP response messages
MyPhoneBean	JavaBean Class to taking information users inputs on TDWS web site
PhoneNumberItems	JavaBean class to taking response
SOAPXMLResponseParser	Using XML parser to parser XML/SOAP part of SOAP response messages

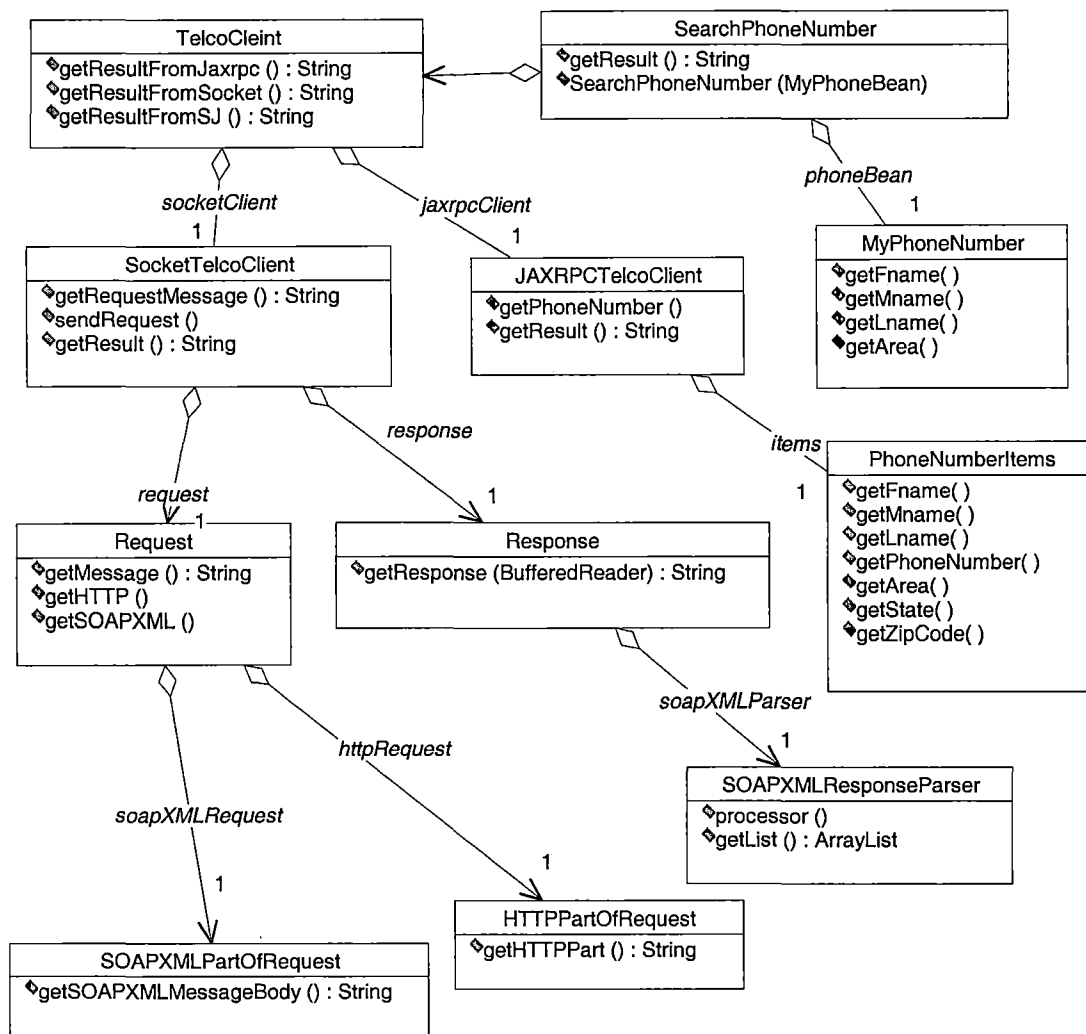


Figure 4.1. Information Aggregator Class Diagram



#### 4.1.2 Information Publisher

4.1.2.1 Telephone Company One Classes Design. The classes in Telco one are summarized in Table 4.3 and class diagram is shown in Figure 4.2.

Table 4.3. Telephone Company One Classes

Class Name	Functions
Telco1WebService	Telco one Web service interface.
RequestHandler	Analysis request and sending response
SOAPXMLRequestParser	Parsing XML/SOAP part of request and get actual request information
DBHandler	Accessing database with request information
Response	Generating SOAP response messages
HTTPPartOfResponse	Generating HTTP part of response
SOAPXMLPartOfResponse	Generating XML/SOAP part of response
DBConnection	Connecting to database system

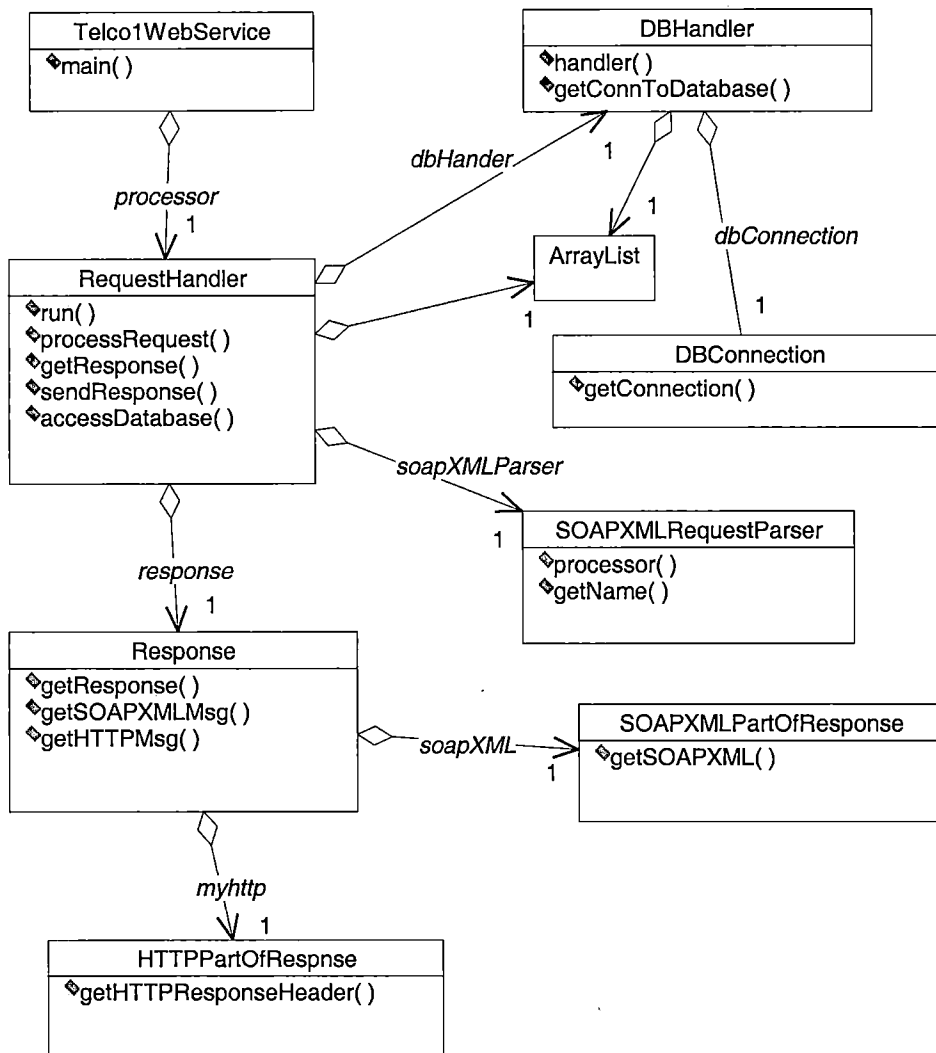


Figure 4.2. Telephone Company One

4.1.2.2 Telephone Company Two Classes Design. Table 4.4 summarizes Telco two classes and Figure 4.3 shows classes' diagram.

Table 4.4. Telephone Company Two Classes

Classes Name	Functions
JAXRPCTelcoIF	Telco Two Web service interface
JAXRPCTelcoImpl	Interface implementation classes
JAXRPCTelcoFetcher	The class is to fetch to JTCWS client
DBHandler	Access database with request information
DBConnection	Connecting to database system

#### 4.1.3 Database

The Information publisher has two telephone companies. Each telephone company has its own DBMS, and the DBMS has two tables: person and phone. Person table stores customers' information, such as name, address, etc. Phone table stores customers' phone numbers. Table 4.5 and Table 4.6 summarize detailed design of table person and phone. Table 4.7 summarizes DBMS classes and Figure 4.4 shows DBMS class diagram.

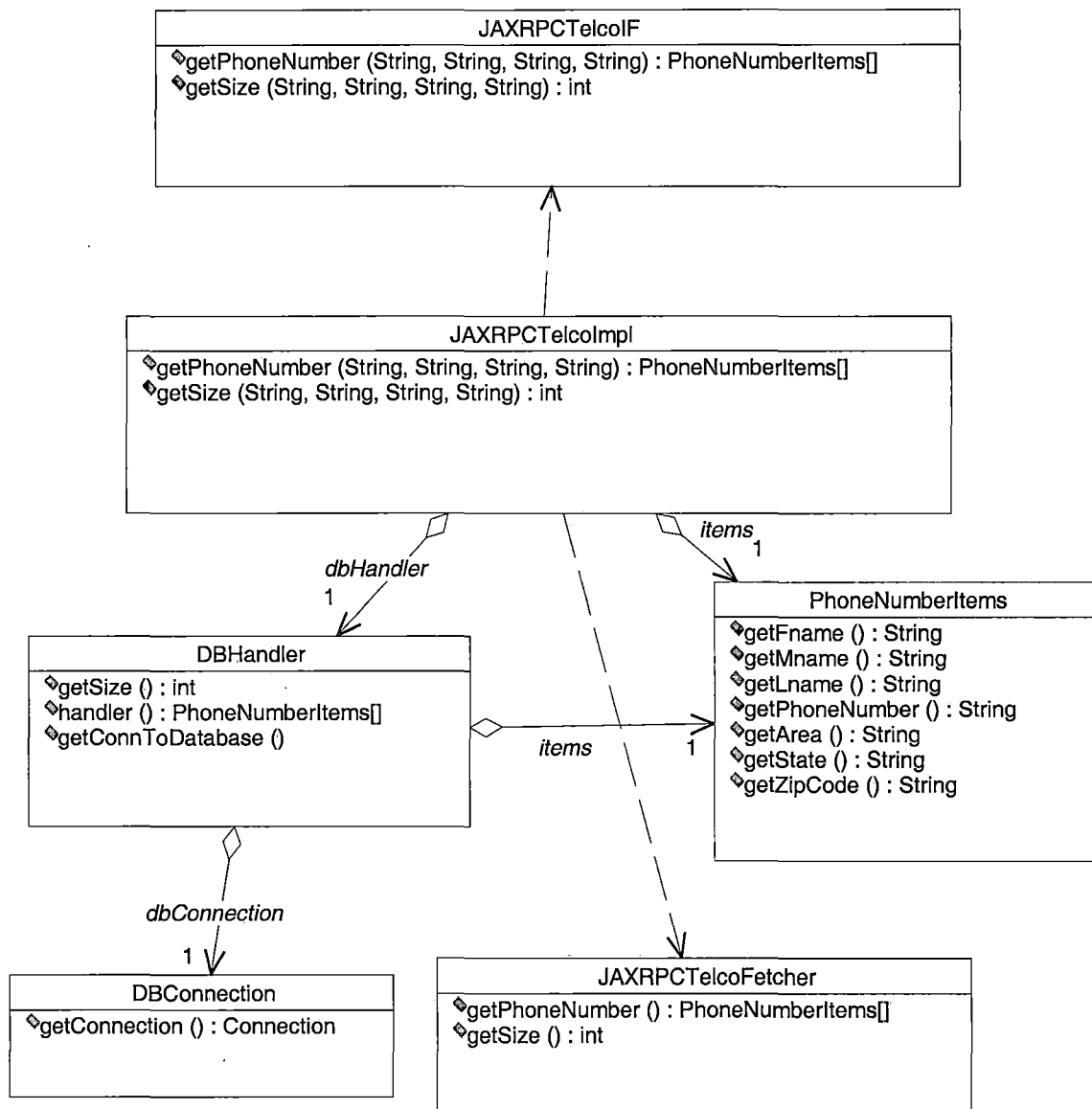


Figure 4.3. Telephone Company Two Class Diagram

Table 4.5. Person Table Design

Entity Name	Type	Description
FirstName	Varchar(80)	NOT NULL
MiddleName	Varchar (80)	
LastName	Varchar (80)	NOT NULL
Street	Varchar (80)	NOT NULL
City	Varchar(80)	NOT NULL
State	Varchar (80)	NOT NULL
ZipCode	Varchar (80)	NOT NULL
Account_id	Serial4	PRIMARY KEY  Serial4 is auto incrementing four-byte integer.

Table 4.6. Phone Table Design

Entity Name	Type	Description
person_id	Int4	REFERENCES person Int4 is four-byte integer.
PhoneNumber	Varchar (80)	PRIMARY KEY

Table 4.7. Database Classes

Classes Name	Functions
CreateTelco	Creating tables and inserting new records and deleting old records
DBConnection	Connecting database system
TelcoBean	Getting information from web sites

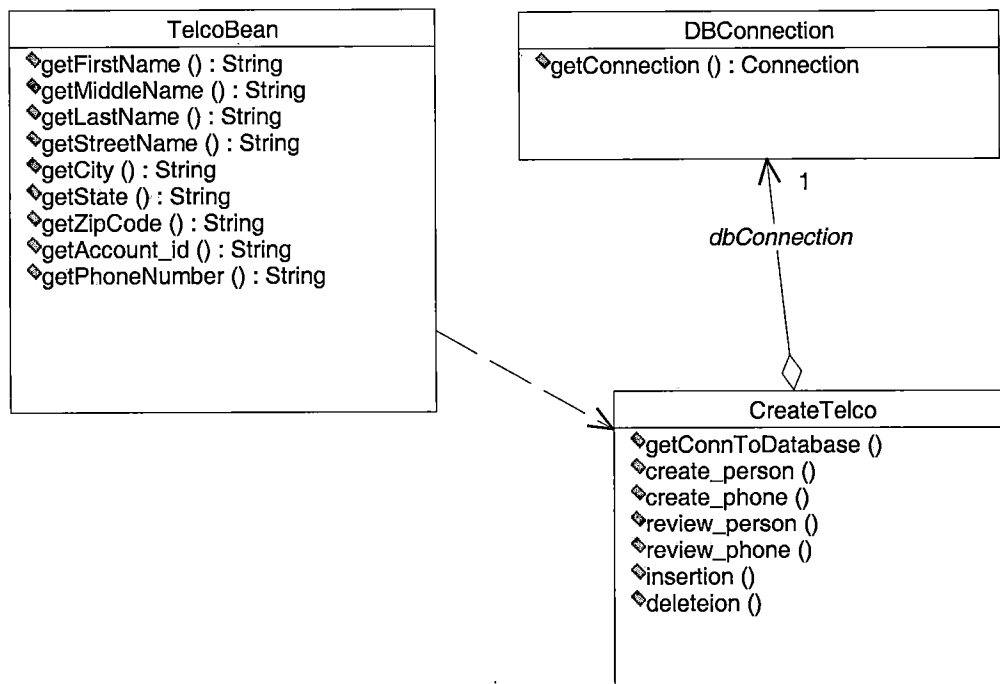


Figure 4.4. Database Class Diagram

## 4.2 Project Classes Implementation

This section shows classes logical algorithms used in this project. To make classes easier to read, all project classes will be written in plain English language.

#### 4.2.1 Information Aggregator

##### 4.2.1.1 The SearchPhoneNumber Class.

```
Class name: SearchPhoneNumber
Function: receives requests from informatin users and
         sends HTML repsonse to infromation users

Begin Class
SearchPhoneNumber

    MyphoneBean  phoneBean
    TelcoClient  telcoClient
    String temp

Constructure SearchPhoneNumber:
Parameter in: MyphoneBean bean
Begin
    phoneBean=bean    //receive request from information users
    telcoClient=new TelcoClient
End;

Function getResult: return value type String
// get response messages
Begin
    if information users specify area City1
        receive response from STCWS
    else if information users specify area City2
        receive response from JTCWS
    else information users don't specify area
        receive reponse from both information publishers
    endif
End;

End Class
```

Figure 4.5. The SearchPhoneNumber Class



4.2.1.2 The TelcoClient Class. The class is shown in Figure 4.6.

```
Class Name: TelcoClient
Functions: Sending rgeusts and receiving responses

Begin Class

TelcoClient
    SocketTelcoClient
    JAXRPCTelcoClient

Function getResultFromJaxrpc: return value type String
// receive response messages from JTCWS
Begin
    Call method in JAXRPCTelcoClient to
        get response messages from JTCWS
End;

Function getResultFromSocket: return value type String
// receive response messages from STCWS
Begin
    Call method in SocketTelcoClient ot
        get response messages from STCWS
End;

Function getResultFromSJ: return value type String
// receive response messages from both information publishers
Begin
    Call method in SocketTelcoClient to
        get response messages from STCWS

    Call method in JAXRPCTelcoClient to
        get response messages from JTCWS
End;

End Class
```

Figure 4.6. The TelcoClient Class

#### 4.2.1.3 The SocketTelcoClient Class. The

SocketTelcoClient class is shown in Figure 4.7.

```
Class Name: SocketTelcoClient
Functions: STCWS client. Sending request and receives response

Begin Class

SocketTelcoClient
    Request request //save SOAP request messages
    Response response //save SOAP response messages

Function getReqestMessage: return value type String
// get SOAP request messages
Begin
    Call method of Request Class
        getMessage
    to get SOAP request messages
End;

Function sendRequest: no return value
// send SOAP request messages to STCWS server
Begin
    write SOAP request messages as Client Socket outputStream
    to Server Socket
End;

Function getResult: return value type String
Parameter in: BufferedReader
// receive HTML response messages
Begin
    Call method of Class Response
        getResponse
    to get HTML response messages
End;

End Class
```

Figure 4.7. The SocketTelcoClient Class

4.2.1.4 The JAXRPCTelcoClient Class. The JAXRPCTelcoClient class is shown in Figure 4.8.

```
Class Name: JAXRPCTelcoClient
Functions: JTCWS client. Making a method call on Stubs and
           receives response from stubs

Begin Class
JAXRPCTelcoClient
    PhoneNumberItems[] items

    // Fetcher of JTCWS client and JTCWS service
    JAXRPCTelcoFetcher fetcher

Function getPhoneNumber: no return value
// receive response from JTCWS service
Begin
    Make a method call on fetcher: getPhoneNumber
    items=return result
End;

Function getResult: return value type String
// generate HTML response messages
Begin
    generate HTML response messages with
        items
End

End Class
```

Figure 4.8. The JAXRPCTelcoClient Class

4.2.1.5 The Request Class. The Request class is shown in Figure 4.9.

```
Class Name: Request
Function: Generating SOAP request messages

Begin Class
Request
    HTTPPartofRequest
    SOAPXMLPartOfRequest

Function: getHTTP: no return value
// get HTTP Header field of SOAP request messages
Begin
    call method in HTTPPartOfRequest class
        getHTTPPart
    to get HTTP header fields of SOAP requests
End;

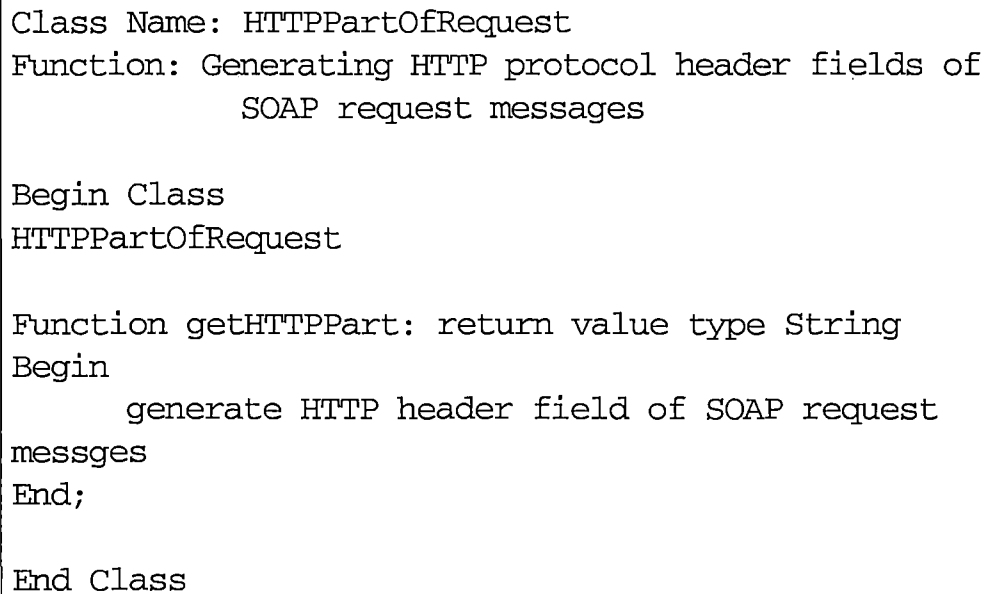
Function: getSOAPXML: no return value
    call method in SOAPXMLPartOfRequest
        getSOAPXMLMessageBody
    to get XML/SOAP part of SOAP requests
End;

Function: getMessage: return value type String
// get SOAP request messages
Begin
    call function getHTTP
    call function getSOAPXML
    generate SOAP request messages
End;

End Class
```

Figure 4.9. The Request Class

4.2.1.6 The HTTPPartOfRequest Class. The HTTPPartOfRequest class is shown in Figure 4.10.



```
Class Name: HTTPPartOfRequest
Function: Generating HTTP protocol header fields of
          SOAP request messages

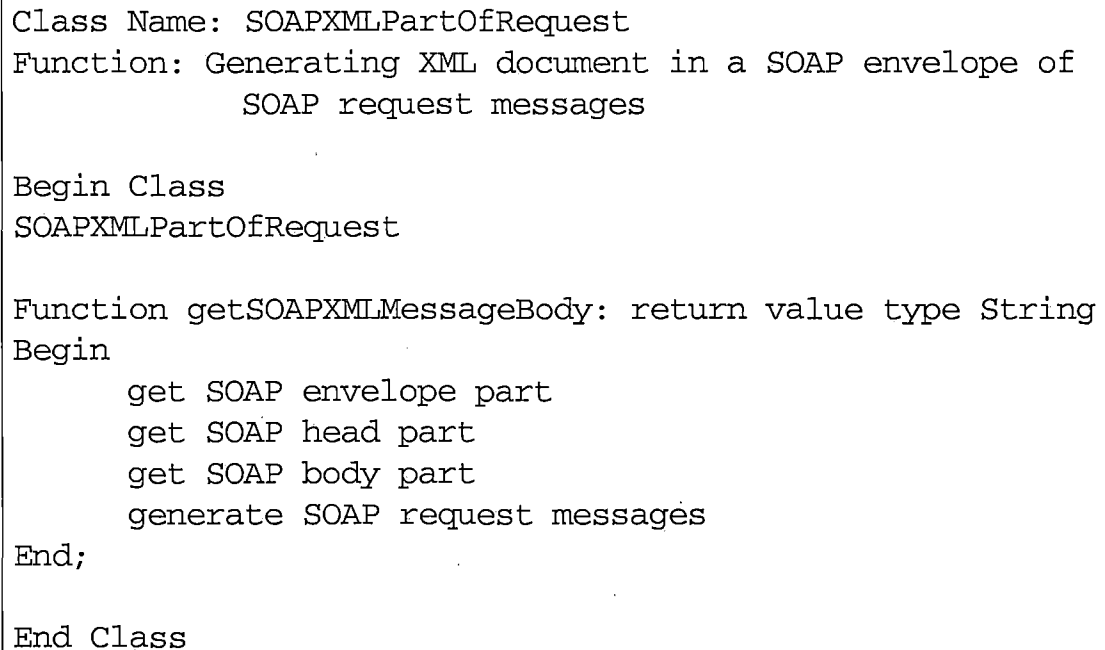
Begin Class
HTTPPartOfRequest

Function getHTTPPart: return value type String
Begin
    generate HTTP header field of SOAP request
messges
End;

End Class
```

Figure 4.10. The HTTPPartOfRequest Class

4.2.1.7 The SOAPXMLPartOfRequest Class. The SOAPXMLPartOfRequest class is shown in Figure 4.11.



```
Class Name: SOAPXMLPartOfRequest
Function: Generating XML document in a SOAP envelope of
          SOAP request messages

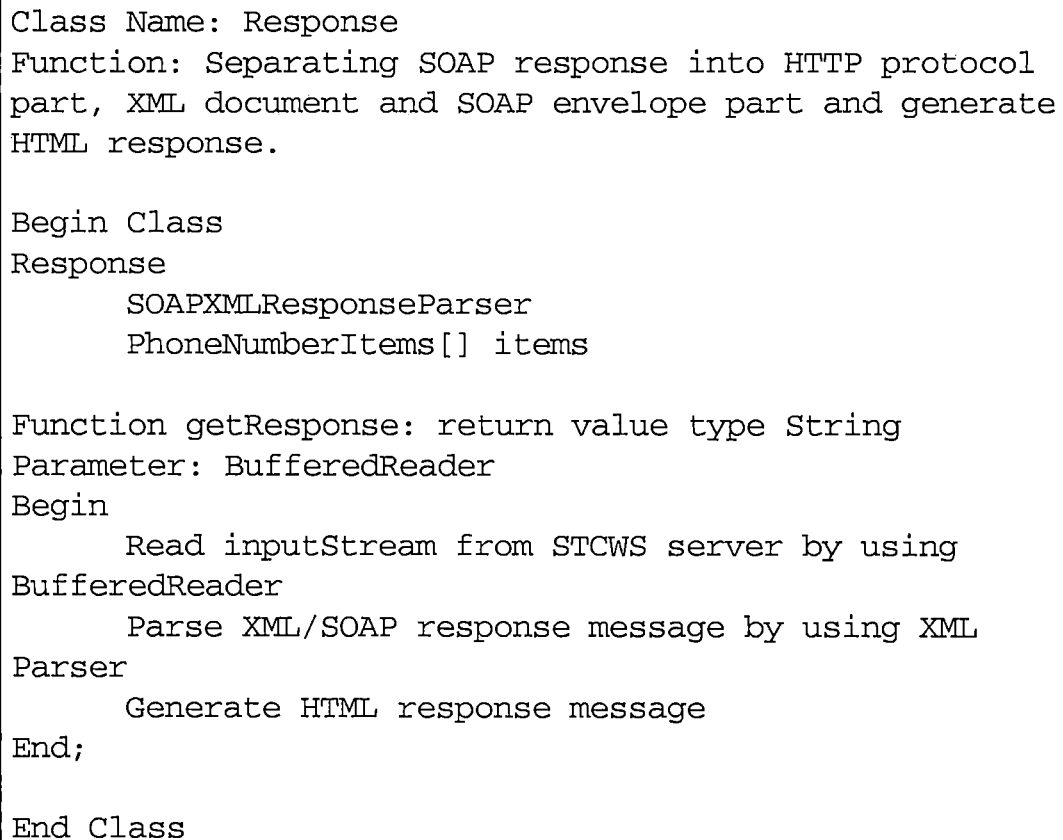
Begin Class
SOAPXMLPartOfRequest

Function getSOAPXMLMessageBody: return value type String
Begin
    get SOAP envelope part
    get SOAP head part
    get SOAP body part
    generate SOAP request messages
End;

End Class
```

Figure 4.11. The SOAPXMLPartOfRequest Class

4.2.1.8 The Response Class. The Response class is shown in Figure 4.12.



```
Class Name: Response
Function: Separating SOAP response into HTTP protocol
part, XML document and SOAP envelope part and generate
HTML response.

Begin Class
Response
    SOAPXMLResponseParser
    PhoneNumberItems[] items

Function getResponse: return value type String
Parameter: BufferedReader
Begin
    Read inputStream from STCWS server by using
BufferedReader
    Parse XML/SOAP response message by using XML
Parser
    Generate HTML response message
End;

End Class
```

Figure 4.12. The Response Class

4.2.1.9 The SOAPXMLResponseParser Class. The SOAPXMLResponseParser class is shown in Figure 4.13.

```
Class Name: SOAPXMLResponseParser
Function: Using XML parser to parser XML/SOAP part of
          SOAP response

Begin Class
SOAPXMLResponseParser
    ArrayList
    PhoneNumberItems[]

Function processor: no return value
Begin
    Read XML message body and
    Extract actual response information and
    Save every phone number into PhoneNumberItems
    as a phone number list
End;

Function getList: return value type ArrayList
Begin
    Save PhoneNumberItems into ArrayList
End;

End Class
```

Figure 4.13. The SOAPXMLResponseParser Class



4.2.1.10 The MyPhoneBean Class. MyPhoneBean class is shown in Figure 4.14.

```
Class Name: MyPhoneNumber
Functions: JavaBean Class to take information users
           request information

Begin Class
MyPhoneNumber
    String lname
    String mname
    String fname
    String area

Function getFname: return value type String
// first name gettre
Begin
    return fname
End;

Function getMname: return value type String
// middle name gettre
Begin
    return mname
End;

Function getLname: return value type String
// last name gettre
Begin
    return lname
End;

Function getArea: return value type String
// area name gettre
Begin
    return area
End;

End Class
```

Figure 4.14. The MyPhoneBean Class

4.2.1.11 The PhoneNumberItems Class. The  
PhoneNumberItems class is shown in Figure 4.15.

```
Class Name: PhoneNumberItems
Function: JavaBean class to taking response informatin
Begin Class
PhoneNumberItems
    String lname
    String mname
    String lname
    String street
    String city
    String state
    String zipCode
    String phoneNumber
Function getFname: return value type String
Begin
    return fname;
End;

Function getMname: return value type String
Begin
    return mname;
End;

Function getLname: return value type String
Begin
    return lname;
End;

Function getStreet: return value type String
Begin
    return street;
End;

Function getCity: return value type String
Begin
    return city;
End;

Function getState: return value type String
Begin
    return state;
End;

Function getZipCode: return value type String
Begin
    return zipCode;
End;

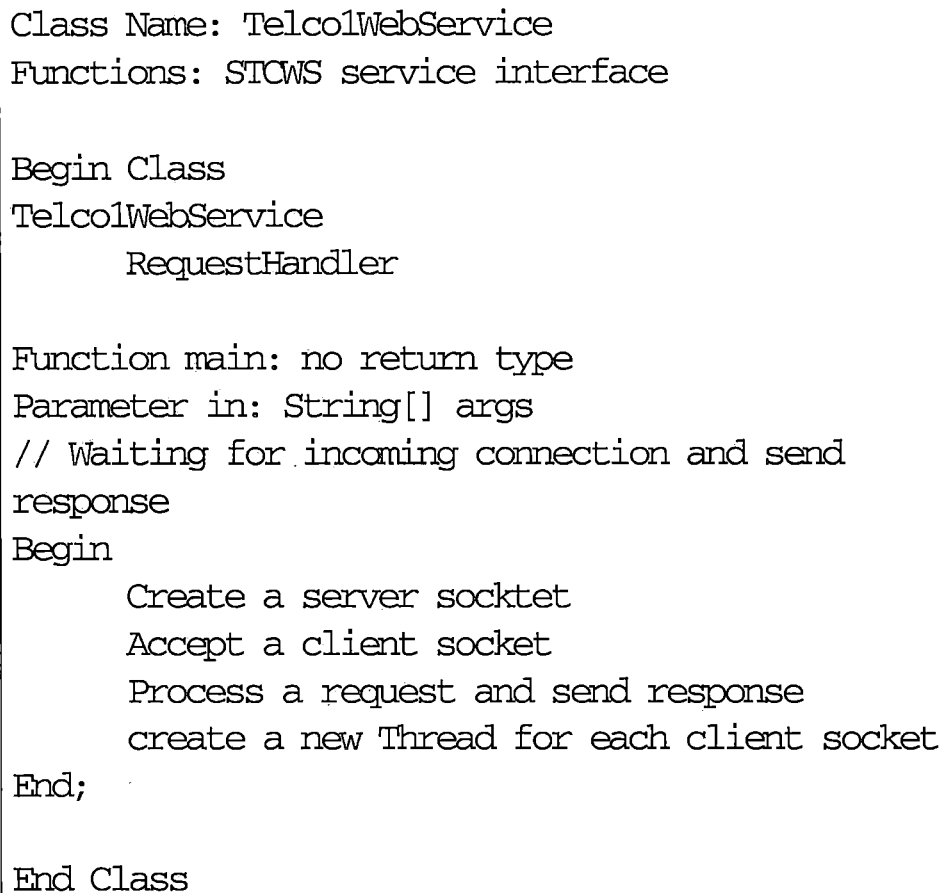
Function getPhoneNumber: return value type String
Begin
    return phoneNumber;
End;
End Class
```

Figure 4.15. The PhoneNumberItems Class

## 4.2.2 Information Publisher

### 4.2.2.1 Telephone Company One.

#### 4.2.2.1.1 The TelcolWebService Class.



```
Class Name: TelcolWebService
Functions: STCWS service interface

Begin Class
TelcolWebService
    RequestHandler

Function main: no return type
Parameter in: String[] args
// Waiting for incoming connection and send
response
Begin
    Create a server socket
    Accept a client socket
    Process a request and send response
    create a new Thread for each client socket
End;

End Class
```

Figure 4.16. The TelcolWebService Class

4.2.2.1.2 The RequestHandler Class. The RequestHandler class is shown in Figure 4.17 and Figure 4.18.

```
Class Name: RequestHandler
Purpose: Analysis request and send response

Begin Class
RequestHandler
    DBHandler
    HTTPRequestParser
    SOAPXMLRequestParser
    Response
    ArrayList
    Hashtable
    String

Function run: no return type
// run method
Begin
    Implement Runnable interface
End;

Function processRequest: no return type
Begin
    While not end of socket inputStream
        Using bufferedReader read SOAP request messages
            as inputStream from client socket
        Get HTTP header field of SOAP request message

        if HTTP header field is end
            Read inputStream and get XML/SOAP request message body
            and write it as an output file
            Call processRequestHelper method

        Break While loop
    End if
    End While
End;
```

Figure 4.17. The RequestHandler Class

```

Function processRequestHelper: no return value
Begin
    Read the output file and
    Call method in SOAPXMLRequestParser class to
        parse XML/SOAP request message body and
    Hashtable=the actual request information

    Call method accessDatabase and
        pass Hashtable as parameter and
        get response information
    ArrayList=the response information

    Call method getResponse and pass ArrayList as
        parameter and get SOAP response messages
    String=SOAP response messages

    Call method sendResponse and pass String as
        parameter
End;

Function accessDatabase: return value type ArrayList
Parameter in: Hashtable
Begin
    Call method in DBHandler class and pass request
        information as parameters and
        get response information
    ArrayList=response information
End;

Function getResponse: return value type String
Parameter in: ArrayList
Begin
    Call method of Response class and pass response
        information as parameters and
        generate SOAP response messages
    String=SOAP response messages
End;

Function sendResponse: no return value
Parameter in: String
Begin
    Write SOAP response message as outputStream to
        Client Socket
End;

End Class

```

Figure 4.18. The RequestHandler Class (Continue)

#### 4.2.2.1.3 The SOAPXMLRequestParser Class.

```
Class Name: SOAPXMLRequestParser
Functions: Parsing XML/SOAP part of request and get actual request
information

Begin Class
SOAPXMLRequestParser
    Hashtable

Function processor: no return type
Begin
    Read XML/SOAP message body elemenet by element
    Save request information into Hashtable
End;

Function getName: return value type Hashtable
Begin
    return Hashtable
End;

End Class
```

Figure 4.19. The SOAPXMLRequestParser Class

#### 4.2.2.1.4 The Response Class.

```
Class Name: Response
Functions: Generating SOAP response messages

Begin Class
Response
    HTTPPartOfResponse
    SOAPXMLPartOfResponse

Function getSOAPXMLMsg: no return type
Begin
    Call method in SOAPXMLPartOfResponse
        getSOAPXML
    to get XML/SOAP message body
End;

Function getHTTPMsg: no return type
Begin
    Call method in HTTPPartOfResponse
        getHTTPResponseHeader
    to get HTTP header field
End;

Function getResponse: return value type String
Begin
    Call method getHTTPMsg and
        getSOAPXMLMsg
    to get SOAP response message
End;

End Class
```

Figure 4.20. The Response Class

#### 4.2.2.1.5 The HTTPPartOfResponse Class.

```
Class Name: HTTPPartOfResponse
Functions: Generating HTTP part of response

Begin Class
HTTPPartOfResponse

Function getHTTPResponseHeader: return value type String
Begin
    generate HTTP part of response message
End;

End Class
```

Figure 4.21. The HTTPPartOfResponse Class



#### 4.2.2.1.6 The SOAPXMLPartOfResponse Class.

```
Class Name: SOAPXMLPartOfResponse
Functions: Generating XML/SOAP part of response

Begin Class
SOAPXMLPartOfResponse

Function getSOAPXML: return value type String
Begin
    get SOAP envelope part
    get SOAP head part
    get SOAP body part

    generate SOAP response messages
End;

End Class
```

Figure 4.22. The SOAPXMLPartOfResponse Class

#### 4.2.2.1.7 The DBHandler Class.

```
Class Name: DBHandler
Functions: Access database with request information and
           get response information

Begin Class
DBHandler
    DBConnection
    ArrayList

Function getConnToDatabase: no return type
// get connection to database
Begin
    Call method in DBConnection
        getConnection
    to connecte to STCWS database system
End;

Function handler: return value type ArrayList
Begin
    Query database with request information
    list=response information
End;

End Class
```

Figure 4.23. The DBHandler Class

#### 4.2.2.1.8 The DBConnection Class.

```
Class Name: DBConnection
Functions: Connection to physical database

Begin Class
DBConnection
    postgresql driver
    url=path of postgresql database

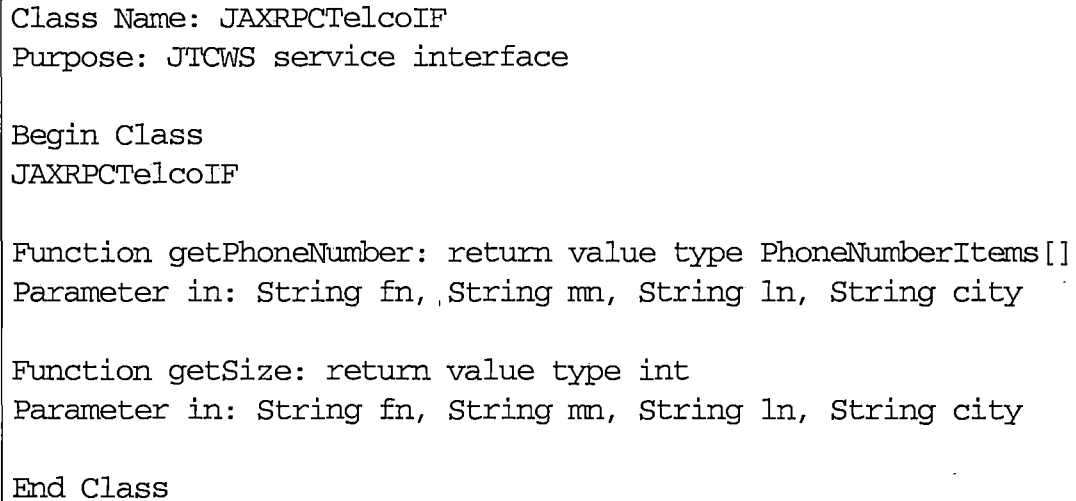
Function getConnection: Connection
// get connection to database
Begin
    Active driver
    Establish connection to STCWS database management system
End;

End Class
```

Figure 4.24. The DBConnection Class

#### 4.2.2.2 Telephone Company Two.

##### 4.2.2.2.1 The JAXRPCTelcoIF Class.



```
Class Name: JAXRPCTelcoIF
Purpose: JTCWS service interface

Begin Class
JAXRPCTelcoIF

Function getPhoneNumber: return value type PhoneNumberItems[]
Parameter in: String fn, String mn, String ln, String city

Function getSize: return value type int
Parameter in: String fn, String mn, String ln, String city

End Class
```

Figure 4.25. The JAXRPCTelcoIF Class

#### 4.2.2.2.2 The JAXRPCTelcoImpl Class.

```
Class Name: JAXRPCTelcoImpl
Purpose: Implementation class Of Interface

Begin Class
JAXRPCTelcoImpl
    DBHandler
    PhoneNumberItems[]
    String size

Function getPhoneNumber: return value type PhoneNumberItems[]
Parameter In: String fn, String mn, String ln, String city
Begin
    Call method of DBHandler class
        getSize
    to get PhoneNumberItems[] array size

    Initialize PhoneNumberItems[]

    Call method of DBHandler
        handler
    to get response information which is saved in PhoneNumberItems[]
    PhoneNumberItems[]=response Information
End;

Function getSize: return value type int
Parameter in: String fn, String mn, String ln, String city
Begin
    Call method of DBHandler
        getSize
    to get size
    size=return size
End;

End Class
```

Figure 4.26. The JAXRPCTelcoImpl Class

#### 4.2.2.2.3 The DBHandler Class.

```
Class Name: DBHandler
Purpose: Access database with request information

Begin Class
DBHandler
    DBConnection
    PhoneNumberItems[]
    String size

Function handler: return value type PhoneNumberItems[]
Begin
    Call method getPhoneNumberItems

    return PhoneNumberItems[]
End;

Function getPhoneNumber: return value type PhoneNumberItems[]
Begin
    Connecte to database

    Query database and save result records into
        PhoneNumberItems[]
    PhoneNumberItems[]=result records

    return PhoneNumberItems[]
End;

Function getSize: return value type int
Begin
    Call method of DBConnection
        getConnection
    to get connection to JTCWS database system

    Query database to get result size
    size=result size

    Close the connection to database
End;

End Class
```

Figure 4.27. The DBHandler Class

#### 4.2.2.2.4 The JAXRPCTelcoFetcher Class.

```
Class Name: JAXRPCTelcoFetcher
Purpose: The class is to fetch to JTCWS client

Begin Class
JAXRPCTelcoFetcher
    PhoneNumberItems[]
    String size

Function getPhoneNumber: return value type PhoneNumberItems[]
Parameter in: String fn, String mn, String ln, String city
Begin
    Create a JAXRPCTelcoIF_Stub object: stub

    Call method of stub class:
        getSize
    to get size to initialize PhoneNumberItems[]
    size=return size

    initialize PhoneNumberItems with size

    Call method of stub class:
        getPhoneNumber with parameter:
            fn, mn, ln, city and
    save response into PhoneNumberItems

    return PhoneNumberItems[]
End;

Function getSize: return value type int
Parameter in: String fn, String mn, String ln, String city
Begin
    Create a JAXRPCTelcoIF_Stub object: stub

    Call method of stub class:
        getSize
    to get size to initialize PhoneNumberItems[]
    size=return size
End;

End Class
```

Figure 4.28. The JAXRPCTelcoFetcher Class

### 4.2.3 Database

#### 4.2.3.1 The CreateTelco Class.

```
Class Name: CreateTelco
Purpose: Creating tables and inserting new records and deleting old records

Begin Class
CreateTelco
    DBConnection
    TelcoBean

Function create_person: no return value
Begin
    Query database to
    Create person table
End;

Function create_phone: no return value
Begin
    Query database to
    Create phone table
End;

Function review_person: no return value
Begin
    Query database to
    Display records saved in person table
End;

Function review_phone: no return value
Begin
    Query database to
    Display records saved in phone table
End;
```

Figure 4.29. The CreateTelco Class



```

Function insertion: no return value
// insert a new record into person and phone tables
Begin
    Query database to
    if the person exists in person table
        if the person's phone number doesn't exist phone table
            Query database and
            insert person's phone number into phone table only
        End if
    else
        Query database and
        insert the new record into person and phone tables
    End if
End;

Function delete_person: no return value
// insert an old record from person and phone tables
Begin
    Query database to
    if the person exists in person table
        Query database and
        delete the record from person and phone table
    End if
End;

Function delete_phone: no return value
// insert an old record from phone tables
Begin
    Query database to
    if the phone number exists in phone table
        Query database and
        delete the record from phone table
    End if
End;

End Class

```

Figure 4.30. The CreatTelco Class (Continue)

#### 4.2.3.2 The TelcoBean Class.

```
Class Name: TelcoBean
Function: JavaBean class to taking response informatin

Begin Class
TelcoBean
    String lastName
    String middleName
    String firstName
    String streetName
    String city
    String state
    String zipCode
    String account_id
    String databaseName
    String phoneNumber

Function getFirstName: return value type String
Begin
    return firstName;
End;

Function getMiddlename: return value type String
Begin
    return middleName;
End;

Function getLastName: return value type String
Begin
    return lastName;
End;

Function getStreetName: return value type String
Begin
    return streetName;
End;
```

Figure 4.31. The TelcoBean Class

```
Function getCity: return value type String
Begin
    return city;
End;

Function getState: return value type String
Begin
    return state;
End;

Function getZipCode: return value type String
Begin
    return zipCode;
End;

Function getPhoneNumber: return value type String
Begin
    return phoneNumber;
End;

Function getAccount_id: return value type String
Begin
    return account_id;
End;

Function getDatabaseName: return value type String
Begin
    return databaseName;
End;

End Class
```

Figure 4.32. The TelcoBean Class (Continue)

## CHAPTER FIVE

### FUTURE ENHANCEMENT AND CONCLUSION

#### 5.1 Future Enhancement

TDWS is a simple Web service application consisting of complicated processes. The TDWS software provides feasibility for further development. The TDWS architecture is described in Chapter Three. From this starting point, TDWS may have additional development.

The Universal Description, Discovery, and Integration (UDDI) is, very generally, a specification that defines registries wherein businesses and Web services (both interfaces and implementations) can be published and found. UDDI is a Web services meeting place. As described in Chapter Two section 2.2, Web service providers can publish their services information into service brokers, and Web service clients can find and get Web services from Web service brokers before they use Web service. Therefore, UDDI is a directory service where businesses can register and search for Web services.

#### 5.2 Conclusion

TDWS is an application of Web Services. It provides the information user an easy way to access public information aggregated from the information publisher through a Web browser.

TDWS also demonstrates the application of Web services on providing convenient and cost-effective access to public telephone directory data. The advantage of the Web service model for the information publisher is that the information publisher can publish its public information with lower cost. This is accomplished by replacing complicated presentation logic within the information aggregator tier, with simple content-only Web Service APIs. The advantage of the web service model for the information aggregator is that the information aggregator can eliminate the need to maintain local data stores, and avoid the problem of serving stale data to the information user. The advantage of the web service model for the information user is that information is more easily obtained through the information aggregator, and the information is not stale.

Since TDWS depends on the ability of parties to communicate with each other on different computing platforms, SOAP request/response messages are developed. In addition, the ability of business entities using different computing platforms to communicate with each other is needed. To realize this ability, Socket paradigm, JAX-RPC technology, and JAXP technology are used.

Moreover, through working on this project, I obtained a lot of experience on using various technologies: Java, JSP, postgresql, JDBC, Tomcat web server, ANT, HTTP protocol,

SOAP protocol, CSS, DTD, XML, HTML, JAX-RPC, JAXP, and JavaScript language.

APPENDIX A  
ACRONYMS AND ABBREVIATIONS

Terminology	Definition
TDWS	Telephone Directory Web Service
STCWS	Socket Telephone Directory Web Service
JTCWS	JAXRPC Telephone Directory web Service
JAX-RPC	Java API for XML-based Remote Procedure all
JAXP	Java API for XML Processing
ANT	Apache ANT. A Java-based build tool.
HTTP	Hypertext Transfer Protocol. The client/server protocol that defines how the messages are formatted and transmitted over World Wild Web
SOAP	Simple Object Access Protocol. SOAP is a simple XML based protocol to let applications exchange information over HTTP.
XML	Extensible Markup Language. XML is a cross-platform, software and hardware independent tool for transmitting information.
Java	An object-oriented language. Java programs are capable of running on most popular computer platforms without the need for recompilation.



Terminology	Definition
Brower	A program capable of retrieving HTML documents that include references to images and Java bit code and rendering it into a user-readable document
API	Application Program Interface. API is the specific method prescribed by a computer operating system, or by another application program, such as Web services.
JSP	JavaServer Page. An extension to the Java servlet technology that provides a simple programming vehicle for displaying dynamic content on a Web page.
JavaScript	A scripting language that is widely supported in Web browsers and other Web tools. It adds interactive functions to HTML pages, which are otherwise static.
JDBC	Java Database Connectivity. A programming interface that lets Java applications access a database via the SQL language.
WSDL	Web Services Description Language. WSDL is an XML-based format for specifying the interface to a Web Service.
UDDI	Universal Discovery, Description and Integration. UDDI is the meeting place for Web Services.

Terminology	Definition
JDK	Java Develop Kit. JDK is Java development program from Sun Microsystems JSDK will be used to compile and run Java programs for this project.
Jakarta Tomcat	Tomcat is JSP server. It will receive JSP file, execute JSP command and response back to client.
Javadoc	Javadoc is the tool from Sun Microsystems for generating API documentation in HTML format from doc comments in source code
Codeviewer	The program package is learned from Dr. David Turner class.

APPENDIX B  
SOURCE CODE

Note: Some setter functions, getter functions, and comments are deleted for saving space.

## Information Aggregator's Source Code:

The build.xml file:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!--
  @Id: build.xml
  @Date: 02/09/2003
  @author: Sun, Hua
-->
<project name="TDWS--Telephone Directories Web Services" default="javadoc" basedir=".">
  <description>
    Telephone Directories Web Services Build File
  </description>
  <property name="CATALINA_HOME" value="/usr/java/jakarta-tomcat-4.1.18"/>
  <property name="JAVA_HOME" value="/usr/java/j2sdk1.4.1_01"/>
  <property name="USER_HOME" value="/home/hsun"/>
  <property name="path" value="${basedir}"/>
  <property name="build" value="${path}/build"/>
  <property name="dist" value="${path}/dist"/>
  <property name="src" value="${path}/src"/>
  <property name="docs" value="${path}/docs"/>
  <property name="application" value="telcoWS_aggregator"/>
  <property name="context_path" value="/${application}"/>
  <property name="telco1" value="${USER_HOME}/public/tdws/publisher/telco1WS"/>
  <property name="telco2" value="${USER_HOME}/public/tdws/publisher/telco2WS"/>
  <property name="common" value="${USER_HOME}/public/tdws/common/build"/>
  <property name="codeviewer" value="${USER_HOME}/public/tdws/codeviewer"/>
  <property name="app_version" value="${application}_1.0"/>
  <property file="${USER_HOME}/build.properties"/>
  <property name="compile_debug" value="true"/>
  <property name="compile_deprecation" value="false"/>
  <property name="compile_optimize" value="true"/>
  <path id="compile.classpath">
    <pathelement location="${CATALINA_HOME}/common/classes"/>
    <fileset dir="${CATALINA_HOME}/common/endorsed">
      <include name="*.jar"/>
    </fileset>
    <fileset dir="${CATALINA_HOME}/common/lib">
      <include name="*.jar"/>
    </fileset>
    <pathelement location="${CATALINA_HOME}/shared/classes"/>
    <fileset dir="${CATALINA_HOME}/shared/lib">
      <include name="*.jar"/>
    </fileset>
    <pathelement location="${telco2}"/>
    <fileset dir="${telco2}/dist">
      <include name="*.jar"/>
    </fileset>
    <pathelement location="${telco1}"/>
    <fileset dir="${telco1}/build/WEB-INF/classes/">
      <include name="**/*.class"/>
    </fileset>
  </path>
  <!-- Initialization target -->
  <target name="init" description="The initialization target">
    <echo message="Creating the time stamp..."/>
    <tstamp/>
  </target>
  <!-- Clean Target -->
  <target name="clean" depends="init"
    description="Delete build and dist directories.">
    <echo message="Deleting old build and dist directories..."/>
    <delete dir="${build}"/>
    <delete dir="${dist}"/>
  </target>
  <!-- Prepare Target -->
```

```

<target name="prepare" depends="clean"
    description="Create build and dist directories.">
    <echo message="Creating build and dist directories..." />
    <mkdir dir="${build}/image"/>
    <mkdir dir="${build}/common"/>
    <mkdir dir="${build}/introduction"/>
    <mkdir dir="${build}/WEB-INF/classes"/>
    <mkdir dir="${build}/WEB-INF/lib"/>
    <mkdir dir="${dist}/docs/api"/>
    <!-- Copy web.xml over from web/WEB-INF -->
    <copy todir="${build}">
        <fileset dir="web">
            <include name="**/*.xml"/>
        </fileset>
    </copy>
    <copy todir="${build}/WEB-INF/lib">
        <fileset dir="web/WEB-INF/lib">
            <include name="*.jar"/>
        </fileset>
        <fileset dir="${telco2}/dist">
            <include name="*.jar"/>
        </fileset>
    </copy>
</target>
<!-- Compile Target -->
<target name="compile" depends="prepare"
    description="Compile a Java Source tree.">
    <echo message="Compiling a Java source tree..." />
    <!-- Copy application resources -->
    <copy todir="${build}/WEB-INF/classes">
        <fileset dir="." excludes="build.xml">
            <include name="soapXMLResponse.xml"/>
        </fileset>
        <fileset dir="${common}" />
    </copy>
    <!-- Compile Java classes as necessary -->
    <javac srcdir="${src}"
        destdir="${build}/WEB-INF/classes"
        debug="${compile_debug}"
        deprecation="${compile_deprecation}"
        optimize="${compile_optimize}"
        fork="yes">
        <include name="**/*.java"/>
        <classpath refid="compile.classpath"/>
        <classpath path="${common}" />
    </javac>
</target>
<!-- Javadoc Target -->
<target name="javadoc" depends="compile"
    description="Generates code documentation using the javadoc tool.">
    <echo message="Generating code documentation using the javadoc tool..." />
    <javadoc destdir="${dist}/docs/api"
        author="true"
        version="true"
        use="true">
        <fileset dir="${src}" defaultexcludes="yes">
            <include name="**/*.java"/>
        </fileset>
        <fileset dir="${telco1}/src" defaultexcludes="yes">
            <include name="**/*.java"/>
        </fileset>
        <fileset dir="${telco2}/src" defaultexcludes="yes">
            <include name="jaxrpc2/*.java"/>
        </fileset>
        <fileset dir="${codeviewer}/src" defaultexcludes="yes">
            <include name="**/*.java"/>
        </fileset>
        <classpath refid="compile.classpath"/>
    </javadoc>
    <copy todir="${build}">
        <fileset dir="${dist}">
            <include name="**/*.xml"/>
        </fileset>
    </copy>
</target>
<!-- Dist Target -->

```

```

    <target name="dist" depends="compile, javadoc" description="Create binary
distribution">
        <echo message="Creating binary distribution..." />
        <!-- Copy documentation subdirectories -->
        <copy todir="${dist}/docs">
            <fileset dir="${docs}" />
        </copy>

        <!-- Create application JAR file -->
        <jar jarfile="${dist}/${application}.war" basedir="${build}" />
    </target>
<!-- Run Target -->
    <target name="run" depends="compile" description="Execute the application">
        <echo message="Executing Telephone Directories Web Services..." />
        <java classpath="${build}/WEB-INF/classes"
            classname="aggregator.SearchPhoneNumber"
            fork="yes" />
    </target>
<!-- All Target -->
    <target name="all" depends="run, dist, debug" description="Just a shortcut">
        <echo message="Executing the web application and creating distribution war file" />
    </target>
<!-- Debug Target -->
    <target name="debug" depends="prepare"
        description="Displays values of some of the properties of this build file">
        <echo message="Displaying values of some of the properties of this build file..." />
        <echo message="USER_HOME= ${USER_HOME}" />
        <echo message="CATALINA_HOME= ${CATALINA_HOME}" />
        <echo message="JAVA_HOME= ${JAVA_HOME}" />
        <echo message="path= ${path}" />
        <echo message="build= ${build}" />
        <echo message="src= ${src}" />
        <echo message="dist= ${dist}" />
        <echo message="docs= ${docs}" />
        <echo message="application= ${application}" />
        <echo message="context_path= ${context_path}" />
        <echo message="classpath= ${build}/WEB-INF/classes" />
        <echo message="username= ${username}" />
        <echo message="password= ${password}" />
    </target>
</project>

```

## The SearchPhoneNumber.java:

```

/** Telephone Directory Web Service
 ** Information Aggregator
 ** Advisor: Dr. David Turner
 ** Author: Hua Sun
 ** Date: Feb. 05, 2003
 **/

package aggregator;

import java.net.*;
import java.io.*;
import java.util.*;
import common.*;

/**
 * Receives a Web request for information user's Web browser
 * and send response to user's browser
 */
public class SearchPhoneNumber
{
    private static MyPhoneBean phoneBean;
    private static String firstName;
    private static String middleName;
    private static String lastName;
    private static String area;
    private static String temp;
    private static TelcoClient telcoClient;
    public SearchPhoneNumber()

```

```

    {
        this.phoneBean = new MyPhoneBean();
        this.firstName = "";
        this.middleName = "";
        this.lastName = "";
        this.area = "";
        this.temp = "";
        this.telcoClient = new TelcoClient();
    }
    public SearchPhoneNumber( MyPhoneBean myPhoneBean )
    {
        this.phoneBean = myPhoneBean;
        this.firstName = myPhoneBean.getFname().trim();
        this.middleName = myPhoneBean.getMname().trim();
        this.lastName = myPhoneBean.getLname().trim();
        this.area = myPhoneBean.getArea().trim();
        this.temp = "";
        this.telcoClient = new TelcoClient( this.firstName,
                                           this.middleName,
                                           this.lastName,
                                           this.area );
    }
    public boolean getCheckJ()
    {
        return telcoClient.getCheckJ();
    }
    public boolean getCheckS()
    {
        return telcoClient.getCheckS();
    }
    public String getResult()
    {
        if( area.equals( "city1" ) )
        {
            temp = getResultS();
        }else if( area.equals( "city2" ) ){
            temp = getResultJ();
        }else if( area.equals( "area" ) ){
            temp = getResultSJ();
        }
        return temp;
    }
    public String getResultSJ()
    {
        if( temp == "" )
        {
            temp = telcoClient.getResultFromSJ();
        }else{
            temp = "";
            temp = telcoClient.getResultFromSJ();
        }
        return temp;
    }
    public static String getResultJ()
    {
        if( temp == "" )
        {
            temp = telcoClient.getResultFromJaxrpc();
        }else{
            temp = "";
            temp = telcoClient.getResultFromJaxrpc();
        }
        return temp;
    }
    public static String getResultS()
    {
        if( temp == "" )
        {
            temp = telcoClient.getResultFromSocket();
        }else{
            temp = "";
            temp = telcoClient.getResultFromSocket();
        }
        return temp;
    }
}

```

## The TelcoClient.java:

```
// Date: Feb. 13, 2003

package aggregator;

import java.io.*;
import java.net.*;
import java.util.*;
/**
 * Send request to a particular telco and get response from the telco
 */
public class TelcoClient
{
    private String fname,
                  mname,
                  lname,
                  area;
    private String temp;
    private boolean checkS;
    private boolean checkJ;
    private SocketTelcoClient socketClient;
    private JAXRPTelcoClient jaxrpcClient;
    public TelcoClient()
    {
        this.temp = "";
        this.fname = "";
        this.mname = "";
        this.lname = "";
        this.area = "";
        this.socketClient = new SocketTelcoClient();
        this.jaxrpcClient = new JAXRPTelcoClient();
    }
    public TelcoClient( String fname,
                       String mname,
                       String lname,
                       String area )
    {
        this.temp = "";
        this.fname = fname;
        this.mname = mname;
        this.lname = lname;
        this.area = area;
        this.checkS = false;
        this.checkJ = false;
    }
    public void sendRequest()
    {
        socketClient.sendRequest();
    }
    public String getResultFromJaxrpc()
    {
        jaxrpcClient = new JAXRPTelcoClient( fname,
                                              mname,
                                              lname,
                                              area );

        if( temp == "" )
        {
            temp = jaxrpcClient.getResult();
        }else{
            temp = "";
            temp = jaxrpcClient.getResult();
        }
        return temp;
    }
    public String getResultFromSocket()
    {
        socketClient = new SocketTelcoClient( fname,
                                              mname,
                                              lname,
                                              area );

        if( temp == "" )
        {
            temp = socketClient.getResult();
        }else{

```



```

        temp = "";
        temp = socketClient.getResult();
    }
    return temp;
}
public boolean getCheckS()
{
    checkS = socketClient.getCheck();
    return checkS;
}
public boolean getCheckJ()
{
    checkJ = jaxrpcClient.getCheck();
    return checkJ;
}
public String getResultFromSJ()
{
    jaxrpcClient = new JAXRPCTelcoClient( fname,
                                           mname,
                                           lname,
                                           area );
    socketClient = new SocketTelcoClient( fname,
                                           mname,
                                           lname,
                                           area );

    String str = socketClient.getResult();
    temp = "";
    if( !getCheckS() )
    {
        if( !getCheckJ() )
        {
            temp = "";
            temp = "Sorry, we were unable to find the person you requested" +
                  "<p> Please check the information and try again </p>" +
                  "<br><br><br>";
        }
        else{
            temp = "";
            temp = jaxrpcClient.getResult();
        }
    }
    else{
        if( getCheckJ() )
        {
            temp = "";
            temp = str;
            temp = temp + jaxrpcClient.getResult();
        }
        else{
            temp = "";
            temp = str;
        }
    }
    return temp;
}
}
}

```

## The Request.java

```

// Date: Feb. 20, 2003
package aggregator;
import java.io.*;
import java.util.*;
/**
 * Generate a SOAP request message
 */
public class Request
{
    private HTTPPartOfRequest httpRequest;
    private SOAPXMLPartOfRequest soapXMLRequest;
    private StringBuffer stringBuffer;
    private int length;
    private String temp;
    public Request()
    {
        this.httpRequest = new HTTPPartOfRequest();
        this.soapXMLRequest = new SOAPXMLPartOfRequest();
    }
}

```

```

        this.stringBuffer = new StringBuffer();
        this.temp = "";
        this.length = 0;
    }
    public Request( String fname, String mname, String lname, String area )
    {
        this.httpRequest = new HTTPPartOfRequest();
        this.soapXMLRequest = new SOAPXMLPartOfRequest( fname, mname, lname, area );
        this.stringBuffer = new StringBuffer();
        this.temp = "";
        this.length = 0;
    }
    public void emit( String line )
    {
        if( stringBuffer == null )
        {
            stringBuffer = new StringBuffer( line );
        }else{
            stringBuffer.append( line );
        }
    }
    public int getLength()
    {
        int l = 0;
        l = soapXMLRequest.getLength();
        return l;
    }
    public void getHTTP()
    {
        length = getLength();
        httpRequest = new HTTPPartOfRequest( length );
        length = 0;
        temp = httpRequest.getHTTPPart();
    }
    public void getSOAPXML()
    {
        temp = temp + soapXMLRequest.getSOAPXMLMessageBody();
    }
    public String getMessage()
    {
        getHTTP();
        getSOAPXML();
        return temp;
    }
}

```

## The HTTPPartOfRequest.java

```

// Date Feb. 18, 2003
package aggregator;
import java.io.*;
import java.util.*;
/**
 * Generate HTTP part of SOAP request messages
 */
public class HTTPPartOfRequest
{
    private String requestLine;
    private String method;
    private String requestURI;
    private String httpVersion;
    private String mimeTypeHeader;
    private String generalHeader;
    private String requestHeader;
    private int size;
    private String length;
    private String host;
    private String accept;
    private String acceptCharset;
    private String connection;
    private String SOAPAction;
    private static String CRLF;
    private String lineEnd;

```

```

public HTTPPartOfRequest()
{
    this.lineEnd = System.getProperty( "line.separator" );
    this.CRLF = "\r\n";
    this.method = "GET";
    this.requestURI = " / ";
    this.httpVersion = "HTTP/1.1";
    this.requestLine = this.method + this.requestURI + this.httpVersion + this.CRLF;
    this.generalHeader = "Transfer-Encoding: chunked" + this.CRLF;
    this.host = "Host: tdws.ias.csusbe.du" + this.CRLF;
    this.accept = "Context-Type: text/soap+xml";
    this.length = "Context-length: nnnn" + this.CRLF;
    this.acceptCharset = "charset=\"iso-8895-1\"" + this.CRLF;
    this.connection = "Connection: close" + this.CRLF;
    this.requestHeader = this.host + this.accept + this.length + this.acceptCharset;
    this.SOAPAction = "http://tdws.ias.csusb.edu" + this.CRLF;
    this.mimeHeader = this.requestHeader + this.SOAPAction + this.CRLF;
}

public HTTPPartOfRequest( int howlong )
{
    this.lineEnd = System.getProperty( "line.separator" );
    this.CRLF = "\r\n";
    this.method = "GET";
    this.requestURI = " / ";
    this.httpVersion = "HTTP/1.1";
    this.requestLine = this.method + this.requestURI + this.httpVersion + this.CRLF;
    this.generalHeader = "Transfer-Encoding: chunked" + this.CRLF;
    this.host = "Host: tdws.ias.csusb.edu" + this.CRLF;
    this.accept = "Context-Type: text/soap+xml; ";
    this.acceptCharset = "charset=\"iso-8895-1\"" + this.CRLF;
    this.size = howlong;
    this.length = "Context-length: " + this.size + this.CRLF;
    this.connection = "Connection: close" + this.CRLF;
    this.requestHeader = this.host + this.accept + this.acceptCharset + this.length;
    this.SOAPAction = "SOAPAction: http://tdws.ias.csusb.edu" + this.CRLF;
    this.mimeHeader = this.requestHeader + this.SOAPAction + this.CRLF;
}

public String getHTTPPart()
{
    String httpPart;
    httpPart = requestLine + mimeHeader;
    return httpPart;
}
}

```

## The SOAPXMLPartOfRequest.java

```

// Date Feb. 18, 2003
package aggregator;
import java.io.*;
import java.util.*;
/**
 * Generate SOAP and XML part of a SOAP request message
 */
public class SOAPXMLPartOfRequest
{
    private String soapENV;
    private String encodingStyle;
    private StringBuffer stringBuffer;

    private String firstName;
    private String middleName;
    private String lastName;
    private String area;
    private String temp;
    private String lineEnd;
    private int length;
    public SOAPXMLPartOfRequest()
    {
        this.soapENV = "http://schemas.xmlsoap.org/soap/envelope/";
        this.encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/";
        this.stringBuffer = new StringBuffer();
    }
}

```

```

        this.temp = "";
        this.firstName = "";
        this.middleName = "";
        this.lastName = "";
        this.area = "";
        this.lineEnd = System.getProperty( "line.separator" );
        this.length = 0;
    }
    public SOAPXMLPartOfRequest( String fname, String mname, String lname, String location
    )
    {
        this.soapENV = "http://schemas.xmlsoap.org/soap/envelope/";
        this.encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/";
        this.stringBuffer = new StringBuffer();
        this.temp = "";
        this.firstName = fname;
        this.middleName = mname;
        this.lastName = lname;
        this.area = location;
        this.lineEnd = System.getProperty( "line.separator" );
        this.length = 0;
    }
    public void envelope()
    {
        temp = "<SOAP-ENV:Envelope " + lineEnd +
            "      xmlns:SOAP-ENV=\"http://schemas.xmlsoap.org/soap/envelope/\" " +
lineEnd +
            "      SOAP-
ENV:encodingStyle=\"http://schemas.xmlsoap.org/soap/encoding/\">" + lineEnd;
    }
    public void header()
    {
        temp = temp +
            "      <SOAP-ENV:Header>" + lineEnd +
            "          <t:Transaction" + lineEnd +
            "              xmlns:t=\"tdws.ias.csusb.edu\" " + lineEnd +
            "              SOAP-ENV:mustUnderstand=\"1\">" + lineEnd +
            "                  5" + lineEnd +
            "          </t:Transaction>" + lineEnd +
            "      </SOAP-ENV:Header>" + lineEnd;
    }
    public void body()
    {
        temp = temp +
            "      <SOAP-ENV:Body>" + lineEnd +
            "          <m:RequestTelephoneNumber
xmlns:m=\"http://tdws.ias.csusb.edu\">" + lineEnd +
            "              <personName>" + lineEnd +
            "                  <firstName>" + firstName + "</firstName>" + lineEnd +
            "                  <middleName>" + middleName + "</middleName>" + lineEnd +
            "                  <lastName>" + lastName + "</lastName>" + lineEnd +
            "                  <location>" + area + "</location>" + lineEnd +
            "              </personName>" + lineEnd +
            "          </m:RequestTelephoneNumber>" + lineEnd +
            "      </SOAP-ENV:Body>" + lineEnd +
            "  </SOAP-ENV:Envelope>" + lineEnd;
    }
    public int getLength()
    {
        int size = 0;
        envelope();
        header();
        body();
        length = temp.length();
        size = length;
        length = 0;
        return size;
    }
}

```

The SOAPXMLResponseParser.java

// Date: Feb. 23, 2003

```

package aggregator;
import java.io.*;
import java.net.*;
import java.util.*;
import javax.xml.parsers.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import common.*;
/**
 * Generate a XML parser and get actual response information
 */
public class SOAPXMLResponseParser extends DefaultHandler
{
    private static SAXParserFactory saxFactory;
    private static SAXParser saxParser;
    private String inputFileName;
    private File inputFile;
    private ArrayList list;
    private PhoneNumberItems items;
    private int i;
    private int size;
    private String noMsg;
    private String temp;
    private boolean yes;
    private boolean no;
    public SOAPXMLResponseParser()
    {
        try{
            this.saxFactory = SAXParserFactory.newInstance();
            this.saxParser = saxFactory.newSAXParser();
            this.inputFileName = "/home/hsun/public/tdws/aggregator/build/WEB-INF/classes/soapXMLResponse.xml";
            this.inputFile = new File( this.inputFileName );
            this.i = 1;
            this.size = 0;
            this.temp = "";
            this.noMsg = "";
            this.list = new ArrayList();
            this.items = new PhoneNumberItems();
            this.yes = false;
            this.no = false;
        } catch( SAXException se ){
            se.printStackTrace();
        } catch( ParserConfigurationException pe ){
            pe.printStackTrace();
        } catch (Throwable t) {
            t.printStackTrace();
        }
    }
    public void processor()
    {
        try{
            saxParser.parse( inputFile, this );
        } catch( SAXException se ){
            se.printStackTrace();
        } catch( IOException e ){
            e.printStackTrace();
        }
    }
    public void startDocument() throws SAXException
    {
    }
    public void endDocument() throws SAXException
    {
        setList( list );
    }
    public void startElement( String uri,
                             String localName,
                             String qName,
                             Attributes attributes )
        throws SAXException
    {
        if( qName.equals( "person" ) )
        {
            no = true;

```

```

    }else if( qName.equals( "person" + i ) ){
        no = true;
    }
    if( no )
    {
        if( qName.equals( "size" ) )
        {
            yes = true;
        }else if( qName.equals( "message" ) ){
            yes = true;
        }else if( qName.equals( "firstName" ) ){
            yes = true;
        }else if( qName.equals( "middleName" ) ){
            yes = true;
        }else if( qName.equals( "lastName" ) ){
            yes = true;
        }else if( qName.equals( "street" ) ){
            yes = true;
        }else if( qName.equals( "city" ) ){
            yes = true;
        }else if( qName.equals( "state" ) ){
            yes = true;
        }else if( qName.equals( "zipCode" ) ){
            yes = true;
        }else if( qName.equals( "phoneNumber" ) ){
            yes = true;
        }
    }
}
}
public void endElement( String uri,
                        String localName,
                        String qName ) throws SAXException
{
    if( yes )
    {
        if( qName.equals( "size" ) )
        {
            size = Integer.parseInt( temp );
        }
        if( no )
        {
            if( qName.equals( "message" ) )
            {
                setNoMsg( temp );
            }else if( qName.equals( "firstName" ) ){
                items.setFname( temp );
            }else if( qName.equals( "middleName" ) ){
                items.setMname( temp );
            }else if( qName.equals( "lastName" ) ){
                items.setLname( temp );
            }else if( qName.equals( "street" ) ){
                items.setStreet( temp );
            }else if( qName.equals( "city" ) ){
                items.setArea( temp );
            }else if( qName.equals( "state" ) ){
                items.setState( temp );
            }else if( qName.equals( "zipCode" ) ){
                items.setZipCode( Integer.parseInt( temp ) );
            }else if( qName.equals( "phoneNumber" ) ){
                items.setPhoneNumber( temp );
                list.add( items );
                if( i < size )
                {
                    items = new PhoneNumberItems();
                    i ++;
                }
                no = false;
            }
        }
    }
    temp = "";
    yes = false;
}
public void characters( char[] ch,
                        int start,
                        int length ) throws SAXException

```

```

    {
        if( yes )
        {
            temp = new String( ch, start, length );
        }
    }
}
public void setNoMsg( String s )
{
    noMsg = s;
}
public String getNoMsg()
{
    return noMsg;
}
public void setList( ArrayList l )
{
    list = l;
}
public ArrayList getList()
{
    return list;
}
public void setItems( PhoneNumberItems i )
{
    items = i;
}
public PhoneNumberItems getItems()
{
    return items;
}
public void setSize( int number )
{
    size = number;
}
public int getSize()
{
    return size;
}
}

```

## The Response.java

```

// Date: Feb. 13, 2003
package aggregator;
import java.io.*;
import java.net.*;
import java.util.*;
import common.*;
/**
 * Generate HTML response for information users
 */
public class Response
{
    private BufferedReader bufferedReader;
    private String outputFileName;
    private File outputFile;
    private FileWriter outputFileWriter;
    private BufferedWriter writer;
    private String lineEnd;
    private String CRLF;
    private String httpHeaderLine;
    private String soapXML;
    private StringBuffer buffer;
    private SOAPXMLResponseParser soapXMLParser;
    private ArrayList list;
    private Iterator iterator;
    private PhoneNumberItems[] items;
    private String responseMsg;
    private int size;
    private boolean check;
    public Response()
    {

```

```

try{
    this.httpHeaderLine = "";
    this.soapXML = "";
    this.responseMsg = "";
    this.lineEnd = System.getProperty( "line.separator" );
    this.outputFileName = "/home/hsun/public/tdws/aggregator/build/WEB-INF/classes/soapXMLResponse.xml";
    this.outputFile = new File( this.outputFileName );
    this.outputFileWriter = new FileWriter( this.outputFile );
    this.writer = new BufferedWriter( this.outputFileWriter );
    this.size = 0;
    this.check = false;
    this.buffer = new StringBuffer();
    this.items = new PhoneNumberItems[ this.size ];
    this.list = new ArrayList();
    this.soapXMLParser = new SOAPXMLResponseParser();
} catch( UnknownHostException uhe ){
    uhe.printStackTrace();
} catch( IOException ioe ){
    ioe.printStackTrace();
}
}

public void emit( String str )
{
    if( buffer == null )
    {
        buffer = new StringBuffer( str );
    } else {
        buffer.append( str );
    }
}

public void readResponse( BufferedReader bufferedReader )
{
    while( true )
    {
        try{
            httpHeaderLine = bufferedReader.readLine();
            if( httpHeaderLine.equals( CRLF ) ||
                httpHeaderLine.equals( " " ) )
            {
                soapXML = bufferedReader.readLine();
                while( soapXML != " " && soapXML.length() != 0 )
                {
                    emit( soapXML );

                    if( soapXML.equals( "</SOAP-ENV:Envelope>" ) )
                    {
                        break;
                    }
                    soapXML = bufferedReader.readLine();
                }
                outputToFile();
                break;
            }
        } catch( IOException ioe ){
            ioe.printStackTrace();
        }
    }
}

public void outputToFile()
{
    try{
        writer.write( buffer.toString() );
        writer.flush();
        writer.close();
    } catch( IOException ioe ){
        ioe.printStackTrace();
    }
}

public void getResponseHelper( BufferedReader br )
{
    readResponse( br );
    soapXMLParser.processor();
    list = soapXMLParser.getList();
    iterator = list.iterator();
    size = list.size();
}

```



```

        items = new PhoneNumberItems[ size ];
        int i = 0;
        try{
            while( iterator.hasNext() )
            {
                if( i < size )
                {
                    items[ i ] = (PhoneNumberItems) iterator.next();
                }
                i ++;
            }
        }catch( NoSuchElementException ne ){
            ne.printStackTrace();
        }
    }
    public boolean getCheck()
    {
        if( size == 0 )
        {
            check = false;
        }else{
            check = true;
        }
        return check;
    }
    public String getResponse( BufferedReader br )
    {
        getResponseHelper( br );
        if( size == 0 )
        {
            responseMsg = "<p>" + getNoMsg() + "</p>" +
                "<p>Check the spelling of name and try again.</p>" +
                "<br><br><br>";
        }else{
            int i = 0;
            while( i < size )
            {
                responseMsg = responseMsg +
                    "<b>Name:</b> " + items[ i ].getFname() + " " +
                    items[ i ].getMname() + " " +
                    items[ i ].getLname() + "<br>" +
                    "<b>Address:</b> " + items[ i ].getStreet() + "<br>" +
                    "<b>City:</b> " + items[ i ].getArea() + ", " +
                    items[ i ].getState() + " " +
                    items[ i ].getZipCode() + "<br>" +
                    "<b>Telephone Number:</b> " + items[ i ].getPhoneNumber() +
                    "<br><br>";
                i ++;
            }
            return responseMsg;
        }
    }
    public String getNoMsg()
    {
        String temp = "";
        temp = soapXMLParser.getNoMsg();
        return temp;
    }
    public PhoneNumberItems[] getItems()
    {
        return items;
    }
    public int getSize()
    {
        return size;
    }
}

```

## The SocketTelcoClient.java

```

// Date: Feb. 13, 2003
package aggregator;
import java.io.*;
import java.net.*;

```

```

import java.util.*;
/**
 * Send a SOAP request to telco one and get a SOAP response from telco one
 */
public class SocketTelcoClient
{
    private Socket socket;
    private InetAddress IPAddress;
    private int port;
    private InputStream inputStream;
    private OutputStream outputStream;
    private InputStreamReader inputStreamReader;
    private BufferedReader bufferedReader;
    private String charsetName;
    private Request request;
    private Response response;
    private String fname, mname, lname, area;
    private String temp;
    private boolean check;
    public SocketTelcoClient()
    {
    }
    public SocketTelcoClient( String fname,
                              String mname,
                              String lname,
                              String area )
    {
        try{
            this.fname = fname;
            this.mname = mname;
            this.lname = lname;
            this.area = area;
            this.port = 9968;
            this.temp = "";
            this.check = false;
            this.IPAddress = InetAddress.getLocalHost();
            this.socket = new Socket( IPAddress, port );
            this.inputStream = this.socket.getInputStream();
            this.inputStreamReader = new InputStreamReader( inputStream );
            this.bufferedReader = new BufferedReader( inputStreamReader );
            this.outputStream = this.socket.getOutputStream();
            this.charsetName = "iso-8859-1";
            this.request = new Request( fname, mname, lname, area );
            this.response = new Response();
        }catch( UnknownHostException uhe ){
            uhe.printStackTrace();
        }catch( IOException ioe ){
            ioe.printStackTrace();
        }
    }
    public String getRequestMessage()
    {
        if( temp == "" )
        {
            temp = request.getMessage();
        }else{
            temp = "";
            temp = request.getMessage();
        }
        return temp;
    }
    public void sendRequest()
    {
        String message = getRequestMessage();
        try{
            outputStream.write( message.getBytes( charsetName ) );
        }catch( UnsupportedEncodingException e ){
            e.printStackTrace();
        }catch( IOException ioe ){
            ioe.printStackTrace();
        }
    }
    public boolean getCheck()
    {
        check = response.getCheck();
    }
}

```

```

        return check;
    }
    public String getResult()
    {
        sendRequest();
        if( temp == "" )
        {
            temp = response.getResponse( bufferedReader );
        }else{
            temp = "";
            temp = response.getResponse( bufferedReader );
        }
        return temp;
    }
    public void close()
    {
        try{
            outputStream.flush();
            outputStream.close();
        }catch( IOException ioe ){
            ioe.printStackTrace();
        }
    }
}

```

## The JAXRPCTelcoClient.java

```

// Date: March 09, 2003
package aggregator;
import java.io.*;
import java.net.*;
import java.util.*;
import javax.xml.rpc.*;
/**
 * Make a method call and get method's response
 */
public class JAXRPCTelcoClient
{
    private PhoneNumberItems[] items;
    private JAXRPCTelcoFetcher fetchers;
    private String fname;
    private String mname;
    private String lname;
    private String area;
    private String temp;
    private String lineEnd;
    private StringBuffer buffer;
    private int size;
    private boolean check;
    public JAXRPCTelcoClient()
    {
        this.size = 0;
        this.items = new PhoneNumberItems[ this.size ];
        this.fname= "";
        this.mname= "";
        this.lname= "";
        this.area = "";
        this.temp = "";
        this.check = false;
        this.lineEnd = System.getProperty( "line.separator" );
        this.buffer = new StringBuffer();
    }
    public JAXRPCTelcoClient( String fn, String mn, String ln, String city )
    {
        this.size = 0;
        this.items = new PhoneNumberItems[ this.size ];
        this.fname= fn;
        this.mname= mn;
        this.lname= ln;
        this.area = city;
        this.temp = "";
        this.check = false;
        this.lineEnd = System.getProperty( "line.separator" );
    }
}

```

```

        this.buffer = new StringBuffer();
    }
    public void emit( String s )
    {
        if( buffer == null )
        {
            buffer = new StringBuffer( s );
        }else{
            buffer.append( s );
        }
    }
    public void getPhoneNumber()
    {
        size = fetchers.getSize( fname, mname, lname, area );
        items = new PhoneNumberItems[ size ];
        items = fetchers.getPhoneNumber( fname, mname, lname, area );
    }
    public boolean getCheck()
    {
        size = fetchers.getSize( fname, mname, lname, area );

        if( size == 0 )
        {
            check = false;
        }else{
            check = true;
        }
        return check;
    }
    public String getResult()
    {
        getPhoneNumber();
        if( size == 0 )
        {
            temp =
                "<p>Sorry, we were unable to find the person you requested in Telephone  

Company Two.</p>" +
                "<p>Check the spelling of name and try again. </p>" +
                "<br><br><br>";
            emit( temp );
            temp = "";
        }else{
            int i = 0;
            while( i < size )
            {
                temp = "<b>Name:</b> " + items[ i ].getFname() + " " +
                    items[ i ].getMname() + " " +
                    items[ i ].getLname() + "<br>" +
                    "<b>Address:</b> " + items[ i ].getStreet() + "<br>" +
                    "<b>City:</b> " + items[ i ].getArea() + ", " +
                    items[ i ].getState() + " " +
                    items[ i ].getZipCode() + "<br>" +
                    "<b>Telephone Number:</b> " + items[ i ].getPhoneNumber() +
                "<br><br>";
                emit( temp );
                temp = "";
                i ++;
            }
            temp = buffer.toString();
            return temp;
        }
    }
}

```

index.jsp:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <title>TDWS-Telephone Directory Web Service--Master Project CSUSB</title>
    <meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
    <script type="text/javascript" language="javascript" src="validation.js">
    </script>
  </head>
  <body topMargin=0>

```

```

<div align=center>
  <table border=0 cellPadding=0 cellSpacing=0 width=760>
    <center>
      <tr bgColor=#ffffff>
        <td height=100 vAlign=center width="100%">
          
        </td>
      </tr>
      <tr>
        <td bgColor=#ffffff width="100%">
          <table bgColor=#000000 border=0 cellPadding=2 cellSpacing=1
width="100%">
            <tr>
              <td align=middle bgColor=#5a67aa width="12%">
                <a class=menutop href="../../home.jsp"><p><font size=3
color=#FFFFFF>TDWS Home</p></font></a>
              </td>
              <td align=middle bgColor=#5a67aa width="12%">
                <a class=menutop href="index.jsp"><p><font size=3
color=#FFFFFF>Search</p></font></a>
              </td>
              <td align=middle bgColor=#5a67aa width="12%">
                <a class=menutop href="../../introduction"><p><font size=3
color=#FFFFFF>About TDWS</p></font></a>
              </td>
            </tr></table></td>
          </tr>
          <tr>
            <td width="100%">
              <div align=center>
                <table border=0 cellPadding=0 cellSpacing=0 width="97%">
                  <tr><td width="100%" height=10></td></tr><tr>
                    <tr><td align=center height=20 width="100%">Search Telephone
Number</td></tr>
                    <tr><td width="100%" height=10></td></tr>
                    <tr>
                      <td align=center width="100%">
                        <p><font size=1 color=#8B0000>( *All fields are required
) </FONT></p></td></tr>
                      <tr><td width="100%" height=30></td></tr>
                      <tr>
                        <td bgColor=#D3D3D3 class=font3 colSpan=1 width="100%">
                          <p><font size=2>Enter Information</font></p>
                        </td></tr>
                        <tr><td width="100%" height=20></td></tr>
                        <tr>
                          <form name=logonForm method=post action="phone.jsp"
onSubmit="return validate()">
                            <td>
                              <table border=0 cellPadding=0 cellSpacing=0
width="100%">
                                <tr><td align=right>First Name:</td>
                                  <td><input type=text name=firstName
size=20></td></tr>
                                <tr><td align=right>Middle Name:</td>
                                  <td><input type=text name=middleName
size=20></td></tr>
                                <tr><td align=right>Last Name*:</td>
                                  <td><input type=text name=lastName
size=20></td></tr>
                                <tr><td align=right>Area:</td>
                                  <td><select name="area" size=1>
                                    <option value="area" selected>area
                                    <option value="city1">City1
                                    <option value="city2">City2
                                  </select></td>
                                </tr>
                                <tr><td height=40></td></tr>
                                <tr>
                                  <td>&nbsp;</td>
                                  <td>
                                    <input type=submit value=Search>
                                    <input type=reset value=Clear>
                                  </td>
                                </tr>
                              </table>
                            </td>
                          </tr>
                        </table>

```

```

        </td>
      </tr>
    </table>
  </div>
</td>
</tr>
</center>
</table>
</div>
</body>
</html>

```

## Phone.jsp:

```

<%@ page language="java" import="aggregator.*, common.*" session="true" %>

<html>
  <head>
    <title>
      TDWS--Telephone Directory Web Service
    </title>
  </head>
  <body>
    <%@ include file="firstname.jsp" %>
    <%@ include file="middlename.jsp" %>
    <%@ include file="lastname.jsp" %>
    <jsp:useBean id="myPhoneBean" class="common.MyPhoneBean"/>
    <jsp:setProperty name="myPhoneBean" property="fname" value="<%= fname%>"/>
    <jsp:setProperty name="myPhoneBean" property="mname" value="<%= mname%>"/>
    <jsp:setProperty name="myPhoneBean" property="lname" value="<%= lname%>"/>
    <jsp:setProperty name="myPhoneBean" property="area" value="<%= area%>"/>

    <div align=center>
      <table border=0 cellPadding=0 cellSpacing=0 width=760>
        <tr bgColor=#ffffff>
          <td height=100 vAlign=center width="100%">
            
          </td>
        </tr>
        <tr>
          <td bgColor=#ffffff width="100%">
            <table border=0 cellPadding=0 cellSpacing=0 width="100%">
              <tr>
                <td width="100%">
                  <table bgColor=#000000 border=0 cellPadding=2 cellSpacing=1
width="100%">
                    <tr>
                      <td align=middle bgColor=#5a67aa width="12%">
                        <a class=menutop href="../home.jsp"><p><font
size=3 color=#FFFFFF>TDWS Home</p></font></a>
                      </td>
                      <td align=middle bgColor=#5a67aa width="12%">
                        <a class=menutop href="index.jsp"><p><font size=3
color=#FFFFFF>Search</p></font></a>
                      </td>
                      <td align=middle bgColor=#5a67aa width="12%">
                        <a class=menutop href="../introduction"><p><font
size=3 color=#FFFFFF>About TDWS</p></font></a>
                      </td>
                    </tr></table></td>
                  </tr>
                <tr>
                  <td width="100%">
                    <div align=center>
                      <table border=0 cellPadding=0 cellSpacing=0 width="100%">
                        <tr>
                          <td bgColor=#ffffff height=40 width="100%">
                        </tr>
                        <tr>
                          <td bgColor=#D3D3D3 class=font3 colSpan=1
width="100%">
                            <p><font size=2>Display Result Information

```

```

        </font></p>
    </td></tr>
    <tr>
        <td bgColor=#ffffff height=20 width="100%">
    </tr>
    <tr>
        <td align=left width="100%">
            <%
                SearchPhoneNumber phone = new
SearchPhoneNumber( myPhoneBean );

                String string = phone.getResult();
                out.print( string );
            %>
        </td>
    </tr>
</table></div>
</td>
</tr>
</table>
</td>
</tr>
</table>
</div>
</body>
</html>

```

The package.xml

```

<body>
This package contains a complete program to provide
telephone directory web site
</body>

```

Telephone Company One Package:

Build.xml

```

<?xml version="1.0" encoding="iso-8859-1" ?>
<!--
    @Id: build.xml
    @Date: 02/29/2003
    @author: Sun, Hua
-->
<project name="TDWS--Telephone Directories Web Services" default="run" basedir=".">
    <description>
        Socket Telephone Web Services Build File
    </description>

    <property name="CATALINA_HOME" value="/usr/java/jakarta-tomcat-4.1.18"/>
    <property name="JAVA_HOME" value="/usr/java/j2sdk1.4.1_01"/>
    <property name="USER_HOME" value="/home/hsun"/>
    <property name="path" value="${basedir}"/>
    <property name="build" value="${path}/build"/>
    <property name="dist" value="${path}/dist"/>
    <property name="src" value="${path}/src"/>
    <property name="docs" value="${path}/docs"/>
    <property name="application" value="telcoWS_telco1WS"/>
    <property name="context_path" value="/${application}"/>
    <property name="common" value="${USER_HOME}/public/tdws/common/build"/>
    <property name="app_version" value="${application}_1.0"/>
    <property file="${USER_HOME}/build.properties"/>
    <property name="compile_debug" value="true"/>
    <property name="compile_deprecation" value="false"/>
    <property name="compile_optimize" value="true"/>

```

```

<path id="compile.classpath">
  <pathelement location="${CATALINA_HOME}/common/classes"/>
  <fileset dir="${CATALINA_HOME}/common/endorsed">
    <include name="*.jar"/>
  </fileset>
  <fileset dir="${CATALINA_HOME}/common/lib">
    <include name="*.jar"/>
  </fileset>
  <pathelement location="${CATALINA_HOME}/shared/classes"/>
  <fileset dir="${CATALINA_HOME}/shared/lib">
    <include name="*.jar"/>
  </fileset>
  <pathelement location="${USER_HOME}/public/tdws/publisher/telcolWS/web/WEB-INF/lib"/>
  <fileset dir="${USER_HOME}/public/tdws/publisher/telcolWS/web/WEB-INF/lib">
    <include name="**/*.jar"/>
  </fileset>
</path>

<!-- Initialization target -->
<target name="init" description="The initialization target">
  <echo message="Creating the time stamp..."/>
  <tstamp/>
</target>

<!-- Clean Target -->
<target name="clean" depends="init"
  description="Delete build and dist directories.">
  <echo message="Deleting old build and dist directories..."/>
  <delete dir="${build}"/>
  <delete dir="${dist}"/>
</target>

<!-- Prepare Target -->
<target name="prepare" depends="clean"
  description="Create build and dist directories.">
  <echo message="Creating build and dist directories..."/>
  <mkdir dir="${build}/WEB-INF/classes"/>
  <mkdir dir="${build}/WEB-INF/lib"/>
  <mkdir dir="${dist}/docs/api"/>

  <!-- Copy application resources -->
  <copy todir="${build}/WEB-INF/classes">
    <fileset dir="${src}" excludes="**/*.java">
      <include name="**/*.xml"/>
    </fileset>
    <fileset dir="." excludes="build.xml">
      <include name="soapXMLRe*.xml"/>
    </fileset>
    <fileset dir="${common}">
    </fileset>
  </copy>

  <!-- Copy web.xml over from web/WEB-INF -->
  <copy todir="${build}/WEB-INF">
    <fileset dir="web/WEB-INF">
      <include name="web.xml"/>
    </fileset>
  </copy>

  <copy todir="${build}">
    <fileset dir="web">
      <include name="**/*.xml"/>
    </fileset>
  </copy>
</target>

<!-- Compile Target -->
<target name="compile" depends="prepare"
  description="Compile a Java Source tree.">
  <echo message="Compiling a Java source tree..."/>

  <javac srcdir="${src}"
    destdir="${build}/WEB-INF/classes"
    debug="${compile_debug}"

```



```

        deprecation="${compile_deprecation}"
        optimize="${compile_optimize}"
        fork="yes">
<include name="**/DBHandler.java"/>
<classpath refid="compile.classpath"/>
<classpath path="${common}"/>
</javac>

<!-- Compile Java classes as necessary -->
<javac srcdir="${src}"
    destdir="${build}/WEB-INF/classes"
    debug="${compile_debug}"
    deprecation="${compile_deprecation}"
    optimize="${compile_optimize}"
    fork="yes">
<include name="**/*.java"/>
<classpath refid="compile.classpath"/>
<classpath path="${common}"/>
</javac>
</target>

<!-- Javadoc Target -->
<target name="javadoc" depends="compile"
    description="Generates code documentation using the javadoc tool.">
<echo message="Generating code documentation using the javadoc tool..."/>
<javadoc destdir="${dist}/docs/api"
    author="true"
    version="true"
    use="true">
<fileset dir="${src}" defaultexcludes="yes">
<include name="**/*.java"/>
</fileset>
<classpath refid="compile.classpath"/>
</javadoc>
</target>

<!-- Dist Target -->
<target name="dist" depends="compile, javadoc" description="Create binary
distribution">
<echo message="Creating binary distribution..."/>

<!-- Copy documentation subdirectories -->
<copy todir="${dist}/docs">
<fileset dir="${docs}"/>
</copy>

<!-- Create application JAR file -->
<jar jarfile="${dist}/${application}.war" basedir="${build}"/>
</target>

<!-- Run Target -->
<target name="run" depends="compile"
    description="Execute the application">
<echo message="Executing Telephone Directories Web Services..."/>
<java classpath="${build}/WEB-INF/classes"
    classname="socket1.TelcolWebService"
    fork="yes"/>
</target>

<!-- All Target -->
<target name="all" depends="run, dist, debug" description="Just a shortcut">
<echo message="Executing the web application and creating distribution war file"/>
</target>

<!-- Debug Target -->
<target name="debug" depends="prepare"
    description="Displays values of some of the properties of this build file">
<echo message="Displaying values of some of the properties of this build file..."/>
<echo message="USER_HOME= ${USER_HOME}"/>
<echo message="CATALINA_HOME= ${CATALINA_HOME}"/>
<echo message="JAVA_HOME= ${JAVA_HOME}"/>
<echo message="path= ${path}"/>
<echo message="build= ${build}"/>
<echo message="src= ${src}"/>
<echo message="dist= ${dist}"/>
<echo message="docs= ${docs}"/>

```

```

        <echo message="application= ${application}"/>
        <echo message="context_path= ${context_path}"/>
        <echo message="classpath= ${build}/WEB-INF/classes"/>
        <echo message="username= ${username}"/>
        <echo message="password= ${password}"/>
    </target>
</project>

```

## SOAPXMLRequestParser.java

// Date: Feb. 23, 2003

```

package socket1;
import java.io.*;
import java.net.*;
import java.util.*;
import javax.xml.parsers.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
/**
 * This class is used to generate a XML parser to parse SOAP request messages.
 */
public class SOAPXMLRequestParser extends DefaultHandler
{
    private static SAXParserFactory saxFactory;
    private static SAXParser saxParser;
    private String inputFileName;
    private File inputFile;
    private Hashtable name;
    private String key;
    private String value;
    private String temp;
    private StringBuffer stringBuffer;
    private boolean yes;
    public SOAPXMLRequestParser()
    {
        try{
            this.saxFactory = SAXParserFactory.newInstance();
            this.saxParser = saxFactory.newSAXParser();
            this.inputFileName = "build/WEB-INF/classes/soapXMLRequest.xml";
            this.inputFile = new File( this.inputFileName );
            this.stringBuffer = new StringBuffer();
            this.name = new Hashtable();
            this.key = "";
            this.value = "";
            this.temp = "";
            this.yes = false;
        } catch( SAXException se ){
            se.printStackTrace();
        } catch( ParserConfigurationException pe ){
            pe.printStackTrace();
        } catch (Throwable t) {
            t.printStackTrace();
        }
    }
    public void processor()
    {
        try{
            saxParser.parse( inputFile, this );
        } catch( SAXException se ){
            se.printStackTrace();
        } catch( IOException e ){
            e.printStackTrace();
        }
    }
    public void emit( String str )
    {
        if( stringBuffer == null )
        {
            stringBuffer = new StringBuffer( str );
        } else{
            stringBuffer.append( str );
        }
    }
}

```

```

    }
    public void startDocument() throws SAXException
    {
    }
    public void endDocument() throws SAXException
    {
        setName( name );
    }
    public void startElement( String uri, String localName, String qName,
        Attributes attributes ) throws SAXException
    {
        if( qName.equals( "firstName" ) )
        {
            setKey( qName );
            yes = true;
        }else if( qName.equals( "middleName" ) ){
            setKey( qName );
            yes = true;
        }else if( qName.equals( "lastName" ) ){
            setKey( qName );
            yes = true;
        }else if( qName.equals( "location" ) ){
            setKey( qName );
            yes = true;
        }
    }
    public void endElement( String uri, String localName, String qName ) throws
SAXException
    {
        if( yes )
        {
            if( temp == "" )
            {
                if( stringBuffer == null )
                {
                    temp = "";
                    setValue( temp );
                }else{
                    temp = temp + stringBuffer;
                    setValue( temp );
                }
            }else{
                temp = "";
                if( stringBuffer == null )
                {
                    temp = "";
                    setValue( temp );
                }else{
                    temp = temp + stringBuffer;
                    setValue( temp );
                }
            }
            name.put( getKey(), getValue() );
        }
        stringBuffer = null;
        yes = false;
    }
    public void characters( char[] ch, int start, int length ) throws SAXException
    {
        if( yes )
        {
            temp = new String( ch, start, length );
            emit( temp );
            temp = "";
        }
    }
    public void setKey( String str )
    {
        if( key == "" )
        {
            key = str;
        }else{
            key = "";
            key = str;
        }
    }
}

```

```

public String getKey()
{
    return key;
}
public void setValue( String str )
{
    if( value == "" )
    {
        value = str;
    }else{
        value = "";
        value = str;
    }
}

public String getValue()
{
    return value;
}
}

```

## RequestHandler.java

// Date: Feb. 20, 2003

```

package socket1;
import java.io.*;
import java.net.*;
import java.util.*;
class RequestHandler implements Runnable
{
    private static Socket socket;
    private InputStream inputStream;
    private InputStreamReader inputStreamReader;
    private BufferedReader bufferedReader;
    private OutputStream outputStream;
    private String CRLF;
    private String lineEnd;
    private String charsetName;
    private StringBuffer soapStringBuffer;
    private String outputFileName;
    private File outputFile;
    private FileWriter outputFileWriter;
    private BufferedWriter writer;
    private String httpHeaderLine;
    private String soapXML;
    private String responseMsg;
    private Hashtable nameTable;
    private ArrayList list;
    private DBHandler dbHandler;
    private HTTPRequestParser httpParser;
    private SOAPXMLRequestParser soapXMLParser;
    private Response response;
    public RequestHandler()
    {
    }
    public RequestHandler( Socket socket )
    {
        try{
            this.socket = socket;
            this.inputStream = this.socket.getInputStream();
            this.inputStreamReader = new InputStreamReader( this.inputStream );
            this.bufferedReader = new BufferedReader( this.inputStreamReader );
            this.outputStream = this.socket.getOutputStream();
            this.httpHeaderLine = "";
            this.soapXML = "";
            this.responseMsg = "";
            this.nameTable = new Hashtable();
            this.CRLF = "\r\n";
            this.lineEnd = System.getProperty( "line.separator" );
            this.charsetName = "iso-8859-1";
            this.soapStringBuffer = new StringBuffer();
            this.outputFileName = "build/WEB-INF/classes/soapXMLRequest.xml";
            this.outputFile = new File( this.outputFileName );

```

```

        this.outputFileWriter = new FileWriter( this.outputFile );
        this.writer = new BufferedWriter( this.outputFileWriter );
        this.list = new ArrayList();
        this.dbHandler = new DBHandler();
        this.httpParser = new HTTPRequestParser();
        this.soapXMLParser = new SOAPXMLRequestParser();
        this.response = new Response();
    }catch( IOException ioe ){
        ioe.printStackTrace();
    }
}

public void main( String args[] ) throws Exception
{
}

public void run()
{
    try{
        processRequest();
    }catch( Exception e ){
        e.printStackTrace();
    }
    finally{
        try{
            socket.close();
        }catch( IOException ioe ){
            ioe.printStackTrace();
        }
    }
}

public void emitSoapXml( String str )
{
    if( soapStringBuffer == null )
    {
        soapStringBuffer = new StringBuffer( str );
    }else{
        soapStringBuffer.append( str );
    }
}

public void processRequest()
{
    System.out.println( "*** Request=" );
    while( true )
    {
        try{
            httpHeaderLine = bufferedReader.readLine();
            System.out.println( httpHeaderLine );
            if( httpHeaderLine.equals( CRLF ) ||
                httpHeaderLine.equals( "" ) )
            {
                soapXML = bufferedReader.readLine();
                System.out.println( soapXML );
                while( soapXML != null )
                {
                    emitSoapXml( soapXML );
                    soapXML = bufferedReader.readLine();
                    System.out.println( soapXML );
                    if( soapXML.equals( "</SOAP-ENV:Envelope>" ) )
                    {
                        emitSoapXml( soapXML );
                        break;
                    }
                }

                processRequestHelper();
                break;
            }else if( httpHeaderLine == null ){
                System.out.println( "The client socket " + socket + " is closed." );
            }
        }catch( IOException ioe ){
            ioe.printStackTrace();
        }
    }
}

public void processRequestHelper()
{
    ArrayList mylist = new ArrayList();

```

```

        getSOAPXML();
        soapXMLParser.processor();
        nameTable = soapXMLParser.getName();
        mylist = accessDatabase( nameTable );
        responsemsg = getResponse( mylist );
        sendResponse( responsemsg );
        System.out.println( "**** RequestHandler processRequestHelper() send response to
client socket." );
    }
    public ArrayList accessDatabase( Hashtable table )
    {
        dbHandler = new DBHandler( table );
        list = dbHandler.handler();
        return list;
    }
    public String getResponse( ArrayList alist )
    {
        String message = "";
        response = new Response( alist );
        message = response.getResponse();
        return message;
    }
    public void close()
    {
        try{
            outputStream.flush();
            outputStream.close();
        }catch( Exception e ){
            e.printStackTrace();
        }
    }
    public void sendResponse( String msg )
    {
        System.out.println( "*** Response=" + msg );
        try{
            outputStream.write( msg.getBytes( charsetName ) );
        }catch( UnsupportedEncodingException e ){
            e.printStackTrace();
        }catch( IOException ioe ){
            ioe.printStackTrace();
        }
    }
    public void getSOAPXML()
    {
        try{
            writer.write( soapStringBuffer.toString() );
            writer.flush();
            writer.close();
        }catch( IOException ioe ){
            ioe.printStackTrace();
        }
    }
    public void setNameTable( Hashtable table )
    {
        nameTable = table;
    }
    public Hashtable getNameTable()
    {
        return nameTable;
    }
}

```

## DBHandler.java

```

// Date: April 4, 2003

package socket1;

import java.io.*;
import java.sql.*;
import java.util.*;
import common.*;
import database.*;

```

```

/**
 * This class is used to query database with request information
 * and get response information
 */
public class DBHandler
{
    private DBConnection dbConnection;
    private Connection connection;
    private ArrayList list;
    private Hashtable table;
    private Enumeration enumeration;
    private PhoneNumberItems items;
    private Statement stm;
    private ResultSet rst;
    private ResultSetMetaData rsmd;

    private String key;
    private String query;
    private String lname;
    private String mname;
    private String fname;
    private String city;
    private String lineEnd;
    private boolean found;
    private int counter;

    public DBHandler()
    {
    }

    public DBHandler( Hashtable table )
    {
        this.dbConnection = new DBConnection( "telcol" );
        this.list = new ArrayList();
        this.table = table;
        this.enumeration = this.table.keys();
        this.key = "";

        while( this.enumeration.hasMoreElements() )
        {
            this.key = (String) this.enumeration.nextElement();

            if( this.key.equals( "firstName" ) )
            {
                this.fname = (String) this.table.get( this.key );
            }
            else if( this.key.equals( "middleName" ) )
            {
                this.mname = (String) this.table.get( this.key );
            }
            else if( this.key.equals( "lastName" ) )
            {
                this.lname = (String) this.table.get( this.key );
            }
            else if( this.key.equals( "location" ) )
            {
                this.city = (String) this.table.get( this.key );
            }
        }

        this.rst = null;
        this.rsmd = null;
        this.query = "";
        this.connection = null;
        this.stm = null;

        this.found = false;
        this.counter = 0;
        this.items = new PhoneNumberItems();
        this.lineEnd = System.getProperty( "line.separator" );
    }

    public void getConnToDatabase()
    {
        try{
            connection = dbConnection.getConnection();
            stm = connection.createStatement();
        }catch( SQLException ex ){
            ex.printStackTrace();
        }catch( Exception exc ){
            exc.printStackTrace();
        }
    }
}

```

```

    }
}

public void getQuery()
{
    query = "SELECT person.firstname, person.middlename, person.lastname,
person.street, person.city, " +
        " person.state, person.zipcode, phone.phonenumber from person, phone " + lineEnd
+
        "WHERE person.account_id=phone.person_id" + lineEnd +
        "AND person.firstname LIKE '%" + fname + "%' " + lineEnd +
        "AND person.middlename LIKE '%" + mname + "%' " + lineEnd +
        "AND person.lastname LIKE '%" + lname + "%' " + lineEnd +
        "ORDER BY person.account_id;" + lineEnd;
}

public ArrayList handler()
{
    ArrayList mylist = new ArrayList();

    getConnToDatabase();
    getQuery();

    mylist = getPhoneNumber();

    return mylist;
}

public void close()
{
    dbConnection.close();
}

public ArrayList getPhoneNumber()
{
    try{
        found = stm.execute( query );
        if( found )
        {
            rst = stm.getResultSet();
            rsmd = rst.getMetaData();
            counter = rsmd.getColumnCount();

            int j = 1;
            while( rst.next() )
            {
                for( int i = 1; i <= counter; i ++ )
                {
                    if( i == j )
                    {
                        items.setFname( rst.getString( i ) );
                    }else if( i == ( j + 1 ) && i <= counter ){
                        if( rst.getString( i ) == null && rst.getString( i ).length() == 0 )
                        {
                            items.setMname( " " );
                        }else{
                            items.setMname( rst.getString( i ) );
                        }
                    }else if( i == ( j + 2 ) && i <= counter ){
                        items.setLname( rst.getString( i ) );
                    }else if( i == ( j + 3 ) && i <= counter ){
                        items.setStreet( rst.getString( i ) );
                    }else if( i == ( j + 4 ) && i <= counter ){
                        items.setArea( rst.getString( i ) );
                    }else if( i == ( j + 5 ) && i <= counter ){
                        items.setState( rst.getString( i ) );
                    }else if( i == ( j + 6 ) && i <= counter ){
                        items.setZipCode( Integer.parseInt( rst.getString( i ) ) );
                    }else if( i == ( j + 7 ) && i <= counter ){
                        items.setPhoneNumber( rst.getString( i ) );
                    }
                }
                list.add( items );
                items = new PhoneNumberItems();
            }
        }
    }
}

```



```

        }catch( SQLException se ){
            se.printStackTrace();
        }
        close();

        return list;
    }
}

```

## TelcolWebService.java

```

// Date: Feb.13, 2003

package socket1;

import java.io.*;
import java.net.*;
import java.util.*;

/** The WebServer class reads a properties file with some initialization
 * settings. The properties file is webServer.properties.
 * @author Hua Sun
 */
public class TelcolWebService
{
    private static ServerSocket telcolServerSocket;
    private static Socket telcolClientSocket;
    private static Thread telcolThread;
    private static RequestHandler processor;

    private static String propertiesFileName;
    private static int portNumber;
    private static String portString;
    private static Properties properties;
    private static File file;
    private static FileInputStream fileInputStream;

    private NoClassDefFoundError error;

    public TelcolWebService()
    {
        try{
            propertiesFileName = "telcolWS.properties";
            properties = new Properties();
            file = new File( propertiesFileName );
            fileInputStream = new FileInputStream( file );

            processor = new RequestHandler();
            error = new NoClassDefFoundError();
        }catch( FileNotFoundException fnfe ){
            fnfe.printStackTrace();
        }catch( NullPointerException npe ){
            npe.printStackTrace();
        }
    }

    public static void main( String args[] ) throws Exception
    {
        TelcolWebService telcolws = new TelcolWebService();

        try{
            properties.load( fileInputStream );

            portString = properties.getProperty( "port" );
            if( portString != null )
            {
                portNumber = Integer.parseInt( portString );
            }

            fileInputStream.close();

            telcolServerSocket = new ServerSocket( portNumber );
            System.out.println( "Telephone Company 1 Web Service is running at " +

```

```

        telcolServerSocket.getInetAddress() + " on " +
        telcolServerSocket.getLocalPort() );

while( true )
{
    telcolClientSocket = telcolServerSocket.accept();

    if( telcolClientSocket == null )
    {
        System.out.println( "Error: null socket created." );
    }else{
        System.out.println( "New client connection accepted " +
            telcolClientSocket.getInetAddress() + ": " +
            telcolClientSocket.getPort() );

        processor = new RequestHandler( telcolClientSocket );

        telcolThread = new Thread( processor );
        telcolThread.start();
    }
}catch( IOException ioe ){
    ioe.printStackTrace();
}
}
}

```

## Response.java

```

// Date Feb. 18, 2003

package socket1;

import java.io.*;
import java.util.*;
import common.*;

/**
 * This class is used to generate a SOAP response message
 */
public class Response
{
    private HTTPPartOfResponse myhttp;
    private SOAPXMLPartOfResponse soapxml;
    private String smessage;
    private String hmessage;
    private String responseMsg;
    private int length;
    private int size;

    public Response()
    {
        this.soapxml = new SOAPXMLPartOfResponse();
        this.myhttp = new HTTPPartOfResponse();
        this.hmessage = "";
        this.smessage = "";
        this.responseMsg = "";
        this.length = 0;
        this.size = 0;
    }

    public Response( ArrayList alist )
    {
        this.soapxml = new SOAPXMLPartOfResponse( alist );
        this.myhttp = new HTTPPartOfResponse();

        this.hmessage = "";
        this.smessage = "";
        this.responseMsg = "";
        this.length = 0;
        this.size = 0;
    }
}

```

```

public int getLength()
{
    length = soapxml.getLength();
    return length;
}

public int getSize()
{
    size = soapxml.getSize();
    return size;
}

public void getSOAPXMLMsg()
{
    smessage = soapxml.getSOAPXML();
}

public void getHTTPMsg()
{
    length = getLength();
    myhttp = new HTTPPartOfResponse( length );
    hmessage = myhttp.getHTTPResponseHeader();
}

public String getResponse()
{
    getHTTPMsg();
    getSOAPXMLMsg();

    responseMsg = hmessage + smessage;
    return responseMsg;
}
}

```

## SOAPXMLPartOfResponse.java

```

// Date Feb. 18, 2003

package socket1;

import java.io.*;
import java.util.*;
import common.*;

/**
 * This class is used to generate SOAP and XML part of response messages
 */
public class SOAPXMLPartOfResponse
{
    private String soapENV;
    private String encodingStyle;
    private ArrayList list;
    private Iterator iterator;
    private PhoneNumberItems[] items;

    private String temp;
    private String lineEnd;

    private int size;
    private int length;

    public SOAPXMLPartOfResponse()
    {
        this.list = new ArrayList();

        this.soapENV = "http://schemas.xmlsoap.org/soap/envelope/";
        this.encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/";

        this.temp = "";
    }
}

```

```

        this.lineEnd = System.getProperty( "line.separator" );
        this.length = 0;

        this.size = 1;
        this.items = new PhoneNumberItems[ this.size ];
    }

    public SOAPXMLPartOfResponse( ArrayList alist )
    {
        this.list = new ArrayList();
        this.list = alist;
        this.size = this.list.size();
        this.items = new PhoneNumberItems[ this.size ];
        this.iterator = this.list.iterator();

        this.soapENV = "http://schemas.xmlsoap.org/soap/envelope/";
        this.encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/";

        this.temp = "";
        this.lineEnd = System.getProperty( "line.separator" );
        this.length = 0;
    }

    public void getRequestInfo()
    {
        int i = 0;

        try{
            while( iterator.hasNext() )
            {
                if( i < size )
                {
                    items[ i ] = (PhoneNumberItems) iterator.next();
                }
                i ++;
            }
        }catch( NoSuchElementException ne ){
            ne.printStackTrace();
        }
    }

    public void envelope()
    {
        temp = "<SOAP-ENV:Envelope " + lineEnd +
            "xmlns:SOAP-ENV=\"http://schemas.xmlsoap.org/soap/envelope/\" " +
lineEnd +
            "SOAP-
ENV:encodingStyle=\"http://schemas.xmlsoap.org/soap/encoding/\">" + lineEnd;
    }

    public void header()
    {
        temp = temp +
            "<SOAP-ENV:Header>" + lineEnd +
            "<t:Transaction" + lineEnd +
            "xmlns:t=\"tdws.ias.csusb.edu\" " + lineEnd +
            "SOAP-ENV:mustUnderstand=\"1\">" + lineEnd +
            "5" + lineEnd +
            "</t:Transaction>" + lineEnd +
            "</SOAP-ENV:Header>" + lineEnd;
    }

    public void body()
    {
        int j = 1;

        temp = temp +
            "<SOAP-ENV:Body>" + lineEnd +
            "<m:ResponseTelephoneNumber xmlns:m=\"http://tdws.ias.csusb.edu\">"
+ lineEnd;

        if( size == 0 )
        {
            temp = temp +
                "<person>" + lineEnd +
                "<size>" + size + "</size>" + lineEnd +

```

```

One." + lineEnd +
    "                <message>Sorry, the person is not in Telephone Company
    "                </message>" + lineEnd +
    "            </person>" + lineEnd;
    }else if( size == 1 ){
        temp = temp +
            "                <person>" + lineEnd +
            "                <size>" + size + "</size>" + lineEnd +
            "                <firstName>" + items[ j-1 ].getFname() +
            "                <middleName>" + items[ j-1 ].getMname() +
            "                <lastName>" + items[ j-1 ].getLname() +
            "                <street>" + items[ j-1 ].getStreet() +
            "                <city>" + items[ j-1 ].getArea() + "</city>" +
            "                <state>" + items[ j-1 ].getState() + "</state>" +
            "                <zipCode>" + items[ j-1 ].getZipCode() +
            "                <phoneNumber>" + items[ j-1 ].getPhoneNumber() +
            "            </person>" + lineEnd;
    }else{
        temp = temp +
            "                <person>" + lineEnd +
            "                <size>" + size + "</size>" + lineEnd;
        while( j <= size )
        {
            temp = temp +
                "                <person" + j + ">" + lineEnd +
                "                <firstName>" + items[ j-1 ].getFname() +
                "                <middleName>" + items[ j-1 ].getMname() +
                "                <lastName>" + items[ j-1 ].getLname() +
                "                <street>" + items[ j-1 ].getStreet() +
                "                <city>" + items[ j-1 ].getArea() + "</city>" +
                "                <state>" + items[ j-1 ].getState() +
                "                <zipCode>" + items[ j-1 ].getZipCode() +
                "                <phoneNumber>" + items[ j-1
            ].getPhoneNumber() + "</phoneNumber>" + lineEnd +
            "            </person>" + j + ">" + lineEnd;

            j ++;
        }

        temp = temp +
            "                </person>" + lineEnd;
    }

    temp = temp +
        "            </m:ResponseTelephoneNumber>" + lineEnd +
        "        </SOAP-ENV:Body>" + lineEnd +
        "    </SOAP-ENV:Envelope>" + lineEnd;
}

public int getLength()
{
    int m = 0;
    String message = "";

    message = getSOAPXML();

    length = message.length();

    m = length;
    length = 0;
}

```

```

        return m;
    }

    public void setSOAPXML( String line )
    {
        temp = line;
    }

    public String getSOAPXML()
    {
        getRequestInfo();

        envelope();
        header();
        body();

        return temp;
    }

    public void setSize( int l )
    {
        size = l;
    }

    public int getSize()
    {
        return size;
    }
}

```

## HTTPPartOfResponse.java

```

// Date: Feb. 23, 2003

package socket1;

import java.io.*;
import java.net.*;
import java.util.*;

/**
 * This class is used to generate HTTP part of SOAP response message
 */
public class HTTPPartOfResponse
{
    private String statusLine;
    private String httpVersion;
    private String statusCode;
    private String reasonPhrase;

    private String generalHeader;

    private String acceptRanges;
    private String ETag;
    private String responseHeader;

    private String allow;
    private String contentEncoding;
    private String contentLanguage;
    private String contentLength;
    private String charsetset;
    private String contentType;
    private String entityHeader;

    private String connection;
    private String SOAPAction;
    private String extensionHeader;

    private static String CRLF;
    private String lineEnd;

    public HTTPPartOfResponse()
    {

```

```

    }

    public HTTPPartOfResponse( int length )
    {
        this.lineEnd = System.getProperty( "line.separator" );
        this.CRLF = "\r\n";

        this.httpVersion = "HTTP/1.1 ";
        this.statusCode = "200 ";
        this.reasonPhrase = "OK";
        this.statusLine = this.httpVersion + this.statusCode + this.reasonPhrase +
this.CRLF;

//        this.generalHeader = "Transfer-Encoding: chunked" + this.CRLF;

//        this.acceptRanges = "Accept-Ranges: bytes" + this.CRLF;
//        this.ETag = "ETag: " + this.CRLF;

        this.responseHeader = this.acceptRanges;

        this.allow = "Allow: GET, HEAD, PUT" + this.CRLF;
//        this.contentEncoding = "Content-Encoding: gzip" + this.CRLF;
        this.contentLanguage = "Content-Language: en" + this.CRLF;
        this.contentLength = "Content-length: " + length + this.CRLF;
        this.charset = "charset=\"iso-8895-1\"";
        this.contentType = "Content-Type: application/soap+xml; " + this.charset +
this.CRLF;
        this.entityHeader = this.allow + this.contentLanguage + this.contentLength +
this.contentType;

        this.connection = "Connection: close" + this.CRLF;
        this.extensionHeader = this.connection;

    }

    public String getHTTPResponseHeader()
    {
        String temp = "";

        temp = statusLine +
//        generalHeader +
//        responseHeader +
        entityHeader +
        extensionHeader + CRLF;

        return temp;
    }
}

```

## Index.jsp

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<%@ page language="java" import="common.*, database.*, java.io.*, java.util.*"
session="true" %>
<html>
<head>
<title>Telephone Company One--TDWS Master Project CSUSB</title>
<meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
<script type="text/javascript" language="javascript" src="validation.js">
</script>
</head>
<body topMargin=0>
<jsp:useBean id="error" scope="request" class="java.util.Vector"/>
<jsp:useBean id="loginbean" scope="request" class="common.TelcoBean"/>
<div align=center>
<table border=0 cellPadding=0 cellSpacing=0 width=760>
<tr bgColor=#FFFFFF>
<td height=110 vAlign=center width="100%">

</td>
</tr>
<tr><td width="100%" height=0></td></tr>
<tr>

```

```

        <td bgColor=#008B8B class=font3 colSpan=1 width="100%">
        <p><font size=2>Login</font></p>
    </td></tr>
    <tr>
        <td>
            <ul>
                <font color="red" size=2>
                    <%
                        if( !error.isEmpty() )
                        {
                            for( int i = 0; i < error.size(); i ++ )
                            {
                                <li><%=error.get( i ) %>
                            }
                            error.removeAllElements();
                        }
                    <%>
                </font>
            </ul>
        </td>
    </tr>
    <tr>
        <form name=logonForm method=post action="checklogin.jsp"
onSubmit="return validate()">
        <td>
            <table border=0 cellPadding=0 cellSpacing=0 width="100%">
                <tr><td align=right>User ID:</td>
                <td><input type=text name=userid size=20
value='<jsp:getProperty name="loginbean"
property="firstName"/>'>
                </td></tr>
                <tr><td align=right>Password:</td>
                <td><input type=password name=password size=20
value='<jsp:getProperty name="loginbean"
property="middleName"/>'>
                </td></tr>
                <tr><td height=40></td></tr>
                <tr>
                    <td>&nbsp;&nbsp;&nbsp;</td>
                    <td>
                        <input type=submit value=Login>
                    </td>
                </tr>
            </table>
        </td>
    </tr>
</form>
</table>
</div>
</body>
</html>

```

## Insert Customer JSP Pages:

### Enter.jsp:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<%@ page language="java" import="common.*, database.*, java.io.*, java.util.*"
session="true" %>
<%@ include file="../../../safe2.jsp" %>

<html>
<head>
    <title>Telephone Company One--TDWS Master Project CSUSB</title>
    <meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
    <script type="text/javascript" language="javascript"
src="../../../common/validation.js">
    </script>
</head>
<body topMargin=0>
    <jsp:useBean id="error" scope="request" class="java.util.Vector"/>
    <jsp:useBean id="telcoBean" scope="request" class="common.TelcoBean"/>

```



```

        </table>
    </div>
</body>
</html>

```

## Validate.jsp:

```

<%@ page language="java" import="common.*, database.*, java.io.*, java.util.*"
session="true" %>
<%@ include file="../../safe2.jsp" %>

<html>
<head>
<title>Telephone Company One--TDWS Master Project CSUSB</title>
<meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
</script>
</head>
<body>
<%@ include file="../../common/firstname.jsp" %>
<%@ include file="../../common/middlename.jsp" %>
<%@ include file="../../common/lastname.jsp" %>
<%@ include file="../../common/middlecode.jsp" %>
<%@ include file="../../common/lastcode.jsp" %>
<%@ include file="../../common/streettwo.jsp" %>
<%@ include file="../../common/streetname.jsp" %>
<%@ include file="../../common/apt.jsp" %>

<jsp:useBean id="error" scope="request" class="java.util.Vector"/>
<jsp:useBean id="telcoBean" scope="request" class="common.TelcoBean"/>
<jsp:setProperty name="telcoBean" property="firstName" value="<%= fname%>"/>
<jsp:setProperty name="telcoBean" property="middleName" value="<%= mname%>"/>
<jsp:setProperty name="telcoBean" property="lastName" value="<%= lname%>"/>
<jsp:setProperty name="telcoBean" property="middlecode" value="<%= middleCode%>"/>
<jsp:setProperty name="telcoBean" property="lastcode" value="<%= lastCode%>"/>
<jsp:setProperty name="telcoBean" property="streetNumber" value="<%= street1%>"/>
<jsp:setProperty name="telcoBean" property="streetName" value="<%= street%>"/>
<jsp:setProperty name="telcoBean" property="apt" value="<%= apt%>"/>
<jsp:setProperty name="telcoBean" property="city" value="<%= area%>"/>
<jsp:setProperty name="telcoBean" property="zipCode" value="<%= zipCode%>"/>
<jsp:setProperty name="telcoBean" property="phoneNumber" value="<%=
phoneNumber%>"/>

<%@ include file="../../common/menu/menubar.jsp" %>
<tr><td width="100%" bgcolor=#D3D3D3 align=left>
<p><font color=#2F4F4F size=2><b>Insert A New
Record</b></font></p></td></tr>
<tr><td height=20></td></tr>
<tr><td align=center height=40 width="100%">
<%
// CreateTable table = new CreateTable();
// table.execute_creation();

try{
    if( mn && yes && (ac && mc && lc) && isapt && iszip && isstr )
    {
        %>
        <jsp:forward page="store.jsp"/>
    </td></tr>
    <tr><td align=center height=40 width="100%">
        <%
    }else{
        if( !mn ){
            error.addElement( "Invalid Middle Name" );
        }
        if(!ac || !mc || !lc )
        {
            error.addElement( "Invalid Phone Number" );
        }
        if( !yes || !isstr || !iszip ){
            error.addElement( "Invalid Street Name" );
        }
        if( !isapt ){
            error.addElement( "Invalid Apartment Number" );
        }
    }
}

```

```

        }
        if( error.size() > 0 )
        {
            <jsp:forward page="enter.jsp"/>
        }else{
            <jsp:forward page="store.jsp"/>
        }
    }
    }catch( IllegalStateException ie ){
        ie.printStackTrace();
    }
    %>
</td></tr>
<tr><td align=center height=30></td></tr>
<tr>
    <td bgColor=#D3D3D3 class=font3 colSpan=1 width="100%">
        <p><font size=2 color=#2F4F4F>
            <b><a href="enter.jsp">Insert More New Records</a></b></font></p>
        </td></tr>
</table>
</div>
</body>
</html>

```

## Confirmation.jsp

```

<%@ page language="java" import="common.*, database.*, java.io.*, java.util.*"
session="true" %>
<%@ include file="../../../safe2.jsp" %>

<html>
    <head>
        <title>Telephone Company One--TDWS Master Project CSUSB</title>
        <meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
        </script>
    </head>
    <body>
        <%@ include file="../../../common/menu/menubar.jsp" %>
        <tr><td width="100%" bgcolor=#D3D3D3 align=left>
            <p><font color=#2F4F4F size=2><b>Insert A New
Record</b></font></p></td></tr>
            <tr><td height=30></td></tr>
            <tr><td align=left height=40 width="100%">
                <%
                    Insertion insert = ( database.Insertion )session.getAttribute( "insert"
                );
                    TelcoBean bean = ( common.TelcoBean )session.getAttribute( "bean" );

                    try{
                        insert.execute_insert();
                        out.println( insert.getMessage());
                    }catch( NullPointerException ne ){
                        ne.printStackTrace();
                    }
                %>
            </td></tr>
            <tr><td width="100%" height=30></td></tr>
            <tr>
                <td bgColor=#D3D3D3 class=font3 colSpan=1 width="100%">
                    <p><font size=2 color=#2F4F4F>
                        <b><a href="enter.jsp">Insert More Records</a></b></font></p>
                    </td></tr>
            </table>
        </div>
    </body>
</html>

```

## Store.jsp

```

<%@ page language="java" import="common.*, database.*, java.io.*, java.util.*"
session="true" %>
<%@ include file="../../../safe2.jsp" %>

```

```

<html>
<head>
<title>Telephone Company One--TDWS Master Project CSUSB</title>
<meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
</script>
</head>
<body>
<jsp:useBean id="telcoBean" scope="request" class="common.TelcoBean"/>
<%@ include file="../../common/menu/menubar.jsp" %>
<tr><td width="100%" bgcolor=#D3D3D3 align=left>
<p><font color=#2F4F4F size=2><b>Insert A New
Record</b></font></p></td></tr>
<tr><td align=left width="100%">
<%
//      CreateTable table = new CreateTable();
//      table.execute_creation();

Insertion insert = new Insertion( telcoBean );
if( session.isNew() )
{
    session.setAttribute( "insert", insert );
    session.setAttribute( "bean", telcoBean );
} else{
    session.setAttribute( "insert", insert );
    session.setAttribute( "bean", telcoBean );
}

if( !insert.getNo() )
{
    %>
</td></tr>
<tr><td height=20></td></tr>
<tr>
<form action="confirmation.jsp">
<td align=center>
<div align=center>
<table border=0 cellPadding=0 cellSpacing=0 width="100%">
<tr><td align=left height=10>
<p><font size=3 color=#2F4F4F>This is the information you
enter:
</font></p></td></tr>
<tr><td align=center height=10></td></tr>
<tr><td align=left height=20>
<p><font size=2 color=#2F4F4F><b>Name:</b>
<%=telcoBean.getFirstName()%> <%=telcoBean.getMiddleName()%> <%= telcoBean.getLastName()
%>
</font></p></td></tr>
<tr><td align=left height=20>
<p><font size=2 color=#2F4F4F><b>Address:</b> <%=
telcoBean.getStreetName() %>
</font></p></td></tr>
<tr><td align=left height=20>
<p><font size=2 color=#2F4F4F><b>City:</b> <%=
telcoBean.getCity() %> <%=telcoBean.getZipCode() %>
</font></p></td></tr>
<tr><td align=left height=20>
<p><font size=2 color=#2F4F4F><b>Telephone Number:</b> <%=
telcoBean.getPhoneNumber() %>
</font></p></td></tr>
<tr><td height=30></td></tr>
<tr>
<td align=left width="50%">
<input type=submit value=OK>
</td>
</tr>
</table></div>
</td>
</form>
</tr>
<tr><td height=10></td></tr>
<tr>
<td width="30%">
<div align=center>
<table border=0 cellPadding=0 cellSpacing=0 width="100%">
<tr>

```

```

        <form action="enter.jsp">
            <td align=center>
                <div align=center>
                    <table border=0 cellpadding=0 cellspacing=0
width="100%">
                        <tr>
                            <td align=left>
                                <input type=submit value=Cancel>
                            </td>
                        </tr>
                    </table></div>
                </td>
            </form>
        </tr></table></div></td>
    </tr>
    <%
        }else{%>
        <tr><td height=15></td></tr>
        <tr><td align=left height=5 width="100%">
            <p>
                <font size=2 color=#2F4F4F>The specified telephone number
<b><%=telcoBean.getPhoneNumber() %></b> is being used by:
                </font></p>
            <%
                out.println( insert.getRecordII() );
            }%>
        </td></tr>
        <tr><td align=left height=20 width="100%">
        </td></tr>
        <tr><td align=center height=20></td></tr>
        <tr>
            <td bgColor=#D3D3D3 class=font3 colSpan=1 width="100%">
                <p><font size=2 color=#2F4F4F>
                    <b><a href="enter.jsp">Insert More New Records</a></b></font></p>
            </td></tr>
        </table>
    </div>
</body>
</html>

```

## Delete customer JSP Pages:

### Index.jsp

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<%@ page language="java" import="common.*, database.*, java.io.*, java.util.*"
session="true" %>
<%@ include file="../../safe2.jsp" %>

<html>
    <head>
        <title>Telephone Company One--TDWS Master Project CSUSB</title>
        <meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
    </head>
    <body topMargin=0>
        <jsp:useBean id="errorMsg" scope="request" class="java.util.Vector"/>
        <jsp:useBean id="telcoBean" scope="request" class="common.TelcoBean"/>

        <%@ include file="../../common/menu/menubar.jsp" %>
        <tr><td align=center height=20 width="100%">
            <p><font color=#2F4F4F>Delete Customer</font></p>
        </td></tr>
        <tr><td width="100%" height=10></td></tr>
        <tr>
            <td bgColor=#D3D3D3 class=font3 colSpan=1 width="100%">
                <p><font size=2 color=#2F4F4F><b>The Record</b></font></p>
            </td></tr>
            <tr><td align=left>
                <ul>
                    <font color="red" size=2>
                        <%
                            for( int i = 0; i < errorMsg.size(); i ++ )
                                {%>

```



```

<jsp:setProperty name="telcoBean" property="middleName" value="<%= mname%>"/>
<jsp:setProperty name="telcoBean" property="lastName" value="<%= lname%>"/>
<jsp:setProperty name="telcoBean" property="middlecode" value="<%= middleCode%>"/>
<jsp:setProperty name="telcoBean" property="lastcode" value="<%= lastCode%>"/>
<jsp:setProperty name="telcoBean" property="phoneNumber" value="<%=
phoneNumber%>"/>

<%@ include file="../../common/menu/menubar.jsp" %>
<tr><td width="100%" bgcolor=#D3D3D3 align=left>
  <p><font color=#2F4F4F size=2><b>Delete Record</b></font></p></td></tr>
<tr>
  <td align=center>
    <div align=center>
      <table border=0 cellpadding=0 cellspacing=0 width="100%">
        <tr><td align=left height=20></td></tr>
        <tr><td align=left height=20>
          <%
            try{
              Deletion del = new Deletion( telcoBean );

              if( session.isNew() )
              {
                session.setAttribute( "del", del );
                session.setAttribute( "searchbean", telcoBean );
              }else{
                session.setAttribute( "del", del );
                session.setAttribute( "searchbean", telcoBean );
              }
              if( del.getYes() )
              {
                <jsp:forward page="list.jsp"/>
              }
            }
          <%>
        </td></tr>
      </table></div>
    </td>
  </tr>
  <tr><td height=30>
    <%
      }else{
        errorMsg.addElement( fname + " " + mname + " " + lname + " with
// telephone number " +
// areaCode + "-" + middleCode + "-" + lastCode + " could not be
found" );
        errorMsg.addElement( "The specified customer could not be found"
);
        if( errorMsg.size() > 0 )
        {
          <%>
        </td></tr>
        <tr><td>
          <jsp:forward page="index.jsp"/>
        </td></tr>
        <tr><td>
          <%
            }else{
              <jsp:forward page="list.jsp"/>
            }
          <%
        </td></tr>
      </table>
    </div>
  </body>
</html>

```

## List.jsp

```

<%@ page language="java" import="common.*, database.*, java.io.*, java.util.*"
session="true" %>
<%@ include file="../../../safe2.jsp" %>

<html>
  <head>
    <title>Telephone Company One--TDWS Master Project CSUSB</title>
    <meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
    </script>
  </head>
  <body>
    <%@ include file="../../../common/menu/menubar.jsp" %>
    <jsp:useBean id="bean" scope="request" class="Common.TelcoBean"/>
    <tr><td width="100%" bgcolor=#D3D3D3 align=left>
      <p><font color=#2F4F4F size=2><b>Delete Record</b></font></p></td></tr>
    <tr><td height=20></td></tr>
    <tr>
      <td align=center>
        <div align=center>
          <table border=1 cellPadding=0 cellSpacing=0 width="100%">
            <tr><th colspan=8>
              <p><font size=2 color=#2F4F4F>The Records to be deleted
              </font></p>
            </th></tr>
            <%
              Deletion del = ( database.Deletion )session.getAttribute(
"del" );
              TelcoBean telcobean = ( common.TelcoBean
)session.getAttribute( "searchbean" );

              if( del.getYes() )
              {
                del.search_person();
                out.println( del.getMessage() );
              }
            %>
          </table></div>
        </td>
      </tr>
    <tr>
      <form name=logonForm method=post action="index.jsp" onSubmit="return">
        <td align=center>
          <div align=center>
            <table border=0 cellPadding=0 cellSpacing=0 width="100%">
              <tr><td height=20></td></tr>
              <tr>
                <td align=center>
                  <input type=submit value=Cancel>
                </td>
              </tr>
            </table></div>
          </td>
        </form>
      </tr>
    <%
      }else{%>
        <p><font color=#2F4F4F size=3><%=telcobean.getFirstName()%>
<%=telcobean.getMiddleName()%> <%=telcobean.getLastName()%> is not in DBMS</font></p>
        <% }
        %>
      <tr><td height=20></td></tr>
      <tr><td width="100%" bgcolor=#D3D3D3 align=left>
        <p><font color=#2F4F4F size=2>
          <a href="index.jsp">Delete More Records</a>
        </font></p></td></tr>
    </table>
  </div>
</body>
</html>

```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<%@ page language="java" import="common.*, database.*, java.io.*, java.util.*"
session="true" %>
<%@ include file="../../../safe2.jsp" %>

<html>
<head>
<title>Telephone Company One--TDWS Master Project CSUSB</title>
<meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
</head>
<body topMargin=0>
<%
try{
String fname = request.getParameter( "firstname" );
String mname = request.getParameter( "middlename" );
String lname = request.getParameter( "lastname" );
String street = request.getParameter( "street" );
String city = request.getParameter( "city" );
String state = request.getParameter( "state" );
String zipcode = request.getParameter( "zipcode" );
String id = request.getParameter( "account_id" );

%>
<jsp:useBean id="recordbean" scope="request" class="common.TelcoBean"/>

<jsp:setProperty name="recordbean" property="firstName" value="<%= fname%>"/>
<jsp:setProperty name="recordbean" property="middleName" value="<%= mname%>"/>
<jsp:setProperty name="recordbean" property="lastName" value="<%= lname%>"/>
<jsp:setProperty name="recordbean" property="streetName" value="<%= street%>"/>
<jsp:setProperty name="recordbean" property="city" value="<%= city%>"/>
<jsp:setProperty name="recordbean" property="state" value="<%= state%>"/>
<jsp:setProperty name="recordbean" property="zipCode" value="<%= zipcode%>"/>
<jsp:setProperty name="recordbean" property="account_id" value="<%= id%>"/>

<%
session.setAttribute( "recordbean", recordbean );
Deletion del = ( database.Deletion )session.getAttribute( "del" );
TelcoBean searchbean = ( common.TelcoBean )session.getAttribute( "searchbean" );
del = new Deletion( recordbean );
session.setAttribute( "del", del );

%>

<%@ include file="../../../common/menu/menubar.jsp" %>
<tr><td align=center height=20 width="100%">
<p><font color=#2F4F4F>Delete Customers</font></p>
</td></tr>
<tr><td width="100%" height=20></td></tr>
<tr>
<td bgColor=#D3D3D3 class=font3 colSpan=1 width="100%">
<p><font size=2 color=#2F4F4F><b>Delete Record</b></font></p>
</td></tr>
<tr><td height=20></td></tr>
<tr><td align=left height=20>
<%
if( del.getNo() )
{
%>
</td></tr>
<tr>
<td align=left>
<jsp:forward page="confirm.jsp"/>
</td>
</tr>
<tr><td align=left>
<%
} else { %>
<font size=2 color=#2F4F4F><b>Name:</b></font>
<font size=2 color=#2F4F4F><%=fname%> <%=mname%> <%=lname%>
</font><br>
<font size=2 color=#2F4F4F><b>Address:</b></font>
<font size=2 color=#2F4F4F><%=street%></font><br>
<font size=2 color=#2F4F4F><b>City:</b></font>
<font size=2 color=#2F4F4F><%=city%>, <%=state%> <%=zipcode%>
</font><br>

```



```

                <p><font size=3 color=#2F4F4F>is not in DBMS</font></p>
            <%
            }
            }catch( Exception e ){
                e.printStackTrace();
            }
            %>
        </td></tr>
    <tr><td height=20></td></tr>
    <tr><td width="100%" bgcolor=#D3D3D3 align=left>
        <p><font color=#2F4F4F size=2>
            <a href="index.jsp">Delete More Records</a>
        </font></p></td></tr>
    </table>
</div>
</body>
</html>

```

## Confirm.jsp

```

<%@ page language="java" import="common.*, database.*, java.util.*" session="true" %>
<%@ include file="../../safe2.jsp" %>

<html>
    <head>
        <title>STCWS-Telephone Company One--TDWS Master Project CSUSB</title>
        <meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
        </script>
    </head>
    <body>
        <%
            TelcoBean recordbean = ( common.TelcoBean )session.getAttribute( "recordbean" );
            TelcoBean searchbean = ( common.TelcoBean )session.getAttribute( "searchbean" );

            String firstname = recordbean.getFirstName();
            String middlename = recordbean.getMiddleName();
            String lastname = recordbean.getLastName();
            String streetname = recordbean.getStreetName();
            String City = recordbean.getCity();
            String State = recordbean.getState();
            String myzip = recordbean.getZipCode();
        %>

        <%@ include file="../../common/menu/menubar.jsp" %>
        <tr><td width="100%" bgcolor=#D3D3D3 align=left>
            <p><font color=#2F4F4F size=2><b>Delete Records</b></font></p></td></tr>
        <tr>
            <form action="delete.jsp">
                <td align=center>
                    <div align=center>
                        <table border=0 cellPadding=0 cellSpacing=0 width="100%">
                            <tr><td height=30></td></tr>
                            <tr><td align=left height=20>
                                <%
                                    Deletion del = ( database.Deletion
                                )session.getAttribute( "del" );

                                if( del.getNo() )
                                {
                                    <p><font color=#2F4F4F size=3>The selected record to be
                                deleted is: </font></p>
                                }
                            </td></tr>
                            <tr><td height=20></td></tr>
                            <tr><td align=left>
                                <div align=center>
                                    <table border=0 cellPadding=0 cellSpacing=0
                                width="100%">
                                    <%
                                        out.println( del.confirm_person() );
                                    %>
                                </table></div>
                            </td></tr>
                            <tr><td height=30></td></tr>
                            <tr>

```

```

        <td align=left>
            <input type=submit value=OK>
        </td>
    </tr>
</table></div>
</td>
</form>
</tr>
<tr>
    <form action="index.jsp">
        <td align=center>
            <div align=center>
                <table border=0 cellpadding=0 cellspacing=0 width="100%">
                    <tr><td height=10></td></tr>
                    <tr>
                        <td align=left>
                            <input type=submit value=Cancel>
                        </td>
                    </tr>
                </table></div>
            </td>
        </form>
    </tr>
    <%
        }else{%>
            <font size=2 color=#2F4F4F><b>Name:</b></font>
            <font size=2 color=#2F4F4F><%=firstname%> <%=middlename%>
<%=lastname%> </font><br>
            <font size=2 color=#2F4F4F><b>Address:</b></font>
            <font size=2 color=#2F4F4F><%=streetname%></font><br>
            <font size=2 color=#2F4F4F><b>City:</b></font>
            <font size=2 color=#2F4F4F><%=City%>, <%=State%> <%=myzip%>
</font><br>
            <p><font size=3 color=#2F4F4F>is not in DBMS</font></p>
        <%
            }
        </td></tr>
    <tr><td height=30></td></tr>
    <tr><td width="100%" bgcolor=#D3D3D3 align=left>
        <p><font color=#2F4F4F size=2>
            <a href="index.jsp">Delete More Records</a>
        </font></p></td></tr>
    </table>
</div>
</body>
</html>

```

## delete.jsp

```

<%@ page language="java" import="common.*, database.*, java.util.*" session="true" %>
<%@ include file="../../../safe2.jsp" %>

<html>
    <head>
        <title>Telephone Company One--TDWS Master Project CSUSB</title>
        <meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
    </head>
    <body>
        <%
            try{
                TelcoBean recordbean = ( common.TelcoBean )session.getAttribute( "recordbean"
            );

            String firstname = recordbean.getFirstName();
            String middlename = recordbean.getMiddlename();
            String lastname = recordbean.getLastName();
            String streetname = recordbean.getStreetName();
            String City = recordbean.getCity();
            String State = recordbean.getState();
            String myzip = recordbean.getZipCode();
            String phone = recordbean.getPhoneNumber();
        <%

```

```

<% include file="../../../common/menu/menubar.jsp" %>
<tr><td width="100%" bgcolor=#D3D3D3 align=left>
  <p><font color=#2F4F4F size=2><b>Delete The Customer's
Record</b></font></p></td></tr>
<tr><td height=30></td></tr>
<tr><td align=left height=40 width="100%">
  <%
    Deletion del = ( database.Deletion )session.getAttribute( "del" );
    if( del.getNo() )
    {
      del.execute_deletionPerson();
    }
  %>
  <p><font color=#2F4F4F size=2>The selected record</font></p>
  <font size=2 color=#2F4F4F><b>Name:</b></font>
  <font size=2 color=#2F4F4F><b>Name:</b></font>
  <%=lastname%> </font><br>
  <font size=2 color=#2F4F4F><b>Address:</b></font>
  <font size=2 color=#2F4F4F><b>Address:</b></font><br>
  <font size=2 color=#2F4F4F><b>City:</b></font>
  <font size=2 color=#2F4F4F><b>City:</b></font>
  <%=lastname%> </font><br>
  <p><font size=3 color=#2F4F4F> has been deleted</font></p>
  <%
    }else{
  %>
  <p><font size=3 color=#2F4F4F>The record </font></p>
  <font size=2 color=#2F4F4F><b>Name: </b></font>
  <font size=2 color=#2F4F4F> <b>Name: </b></font>
  <%=lastname%></font><br>
  <font size=2 color=#2F4F4F><b>Street:</b></font>
  <font size=2 color=#2F4F4F> <b>Street:</b></font><br>
  <font size=2 color=#2F4F4F><b>City:</b></font>
  <font size=2 color=#2F4F4F> <b>City:</b></font>
  <%=lastname%></font><br>
  <p><font size=3 color=#2F4F4F>is not in DBMS</font></p>
  <%
    }
  %>
  <td></td>
  <td height=20></td></tr>
<tr><td width="100%" bgcolor=#D3D3D3 align=left>
  <p><font color=#2F4F4F size=2>
    <a href="index.jsp">Delete More Records</a>
  </font></p></td></tr>
</table>
</div>
</body>
</html>

```

## Edit Phone Number JSP Pages:

### Index.jsp

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<%@ page language="java" import="common.*, database.*, java.io.*, java.util.*"
session="true" %>
<% include file="../../../safe2.jsp" %>

<html>
  <head>
    <title>Telephone Company One--TDWS Master Project CSUSB</title>
    <meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
  </head>
  <body topMargin=0>
    <jsp:useBean id="errorMsg" scope="request" class="java.util.Vector"/>
    <jsp:useBean id="telcoBean" scope="request" class="common.TelcoBean"/>

    <% include file="../../../common/menu/menubar.jsp" %>
    <tr><td align=center height=20 width="100%">
      <p><font color=#2F4F4F>Edit Telephone Number</font></p>

```





```

    </div>
</body>
</html>
List.jsp

```

```

<%@ page language="java" import="common.*, database.*, java.io.*, java.util.*"
session="true" %>
<%@ include file="../../safe2.jsp" %>

<html>
  <head>
    <title>Telephone Company One--TDWS Master Project CSUSB</title>
    <meta content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
  </head>
  <body>
    <%@ include file="../../common/menu/menubar.jsp" %>
    <jsp:useBean id="bean" scope="request" class="common.TelcoBean"/>
    <tr><td width="100%" bgcolor=#D3D3D3 align=left>
      <p><font color=#2F4F4F size=2><b>Update Records</b></font></p></td></tr>
    <tr><td height=20></td></tr>
    <tr>
      <td align=center>
        <div align=center>
          <table border=1 cellpadding=0 cellspacing=0 width="100%">
            <tr><th colspan=8>
              <p><font size=2 color=#2F4F4F>The Records to be edited
              </font></p>
            </th></tr>
            <%
              UpdatePhone upd = ( database.UpdatePhone
)session.getAttribute( "upd" );
              TelcoBean telcobebean = ( common.TelcoBean
)session.getAttribute( "searchbean" );

              if( upd.getYes() )
              {
                upd.search_person();
                out.println( upd.getMessage() );
              }
            %>
          </table></div>
        </td>
      </tr>
    <tr>
      <td>
        <form name=logonForm method=post action="index.jsp" onSubmit="return">
          <td align=center>
            <div align=center>
              <table border=0 cellpadding=0 cellspacing=0 width="100%">
                <tr><td height=20></td></tr>
                <tr>
                  <td align=center>
                    <input type=submit value=Cancel>
                  </td>
                </tr>
              </table></div>
            </td>
          </form>
        </tr>
      <tr><td>
        <%
          }else{%>
            <p><font color=#2F4F4F size=3><%=telcobebean.getFirstName()%>
<%=telcobebean.getMiddleName()%> <%=telcobebean.getLastName()%> is not in DBMS</font></p>
            <%
              }
            %>
          </td></tr>
        <tr><td height=20></td></tr>
        <tr><td width="100%" bgcolor=#D3D3D3 align=left>
          <p><font color=#2F4F4F size=2>
            <a href="index.jsp">Edit More Records</a>
          </font></p></td></tr>
        </table>
      </div>
    </body>
  </html>

```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<%@ page language="java" import="common.*, database.*, java.io.*, java.util.*"
session="true" %>
<%@ include file="../../safe2.jsp" %>

<html>
<head>
<title>Telephone Company One--TDWS Master Project CSUSB</title>
<meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
</head>
<body topMargin=0>
<%
    try{
        String fname = request.getParameter( "firstname" );
        String mname = request.getParameter( "middlename" );
        String lname = request.getParameter( "lastname" );
        String street = request.getParameter( "street" );
        String city = request.getParameter( "city" );
        String state = request.getParameter( "state" );
        String zipcode = request.getParameter( "zipcode" );
        String phone = request.getParameter( "phonenumber" );
        String id = request.getParameter( "account_id" );
    %>
    <jsp:useBean id="recordbean" scope="request" class="common.TelcoBean"/>

    <jsp:setProperty name="recordbean" property="firstName" value="<%= fname%>"/>
    <jsp:setProperty name="recordbean" property="middleName" value="<%= mname%>"/>
    <jsp:setProperty name="recordbean" property="lastName" value="<%= lname%>"/>
    <jsp:setProperty name="recordbean" property="streetName" value="<%= street%>"/>
    <jsp:setProperty name="recordbean" property="city" value="<%= city%>"/>
    <jsp:setProperty name="recordbean" property="state" value="<%= state%>"/>
    <jsp:setProperty name="recordbean" property="zipCode" value="<%= zipcode%>"/>
    <jsp:setProperty name="recordbean" property="phoneNumber" value="<%= phone%>"/>
    <jsp:setProperty name="recordbean" property="account_id" value="<%= id%>"/>

    <%
        session.setAttribute( "recordbean", recordbean );
        UpdatePhone upd = ( database.UpdatePhone )session.getAttribute( "upd" );
        upd = new UpdatePhone( recordbean );
        session.setAttribute( "upd", upd );
    %>

    <%@ include file="../../common/menu/menubar.jsp" %>
    <tr><td align=center height=20 width="100%">
        <p><font color=#2F4F4F>Edit Records</font></p>
    </td></tr>
    <tr><td width="100%" height=20></td></tr>
    <tr>
        <td bgColor=#D3D3D3 class=font3 colSpan=1 width="100%">
            <p><font size=2 color=#2F4F4F><b>The Record</b></font></p>
        </td></tr>
    <tr><td height=20></td></tr>
    <tr><td align=left height=20>
        <%
            if( upd.getNo() )
            {
                %>
            </td></tr>
            <tr>
                <td align=left>
                    <jsp:forward page="phonenumber.jsp"/>
                </td>
            </tr>
            <tr><td align=left>
                <%
            }else{%>
                <font size=2 color=#2F4F4F><b>Name:</b></font>
                <font size=2 color=#2F4F4F><%=fname%> <%=mname%> <%=lname%>
</font><br>
                <font size=2 color=#2F4F4F><b>Address:</b></font>
                <font size=2 color=#2F4F4F><%=street%></font><br>
                <font size=2 color=#2F4F4F><b>City:</b></font>

```

```

        <font size=2 color=#2F4F4F><%=city%>, <%=state%> <%=zipcode%>
</font><br>
        <font size=2 color=#2F4F4F><b>Phone Number:</b></font>
        <font size=2 color=#2F4F4F><%=phone%></font><br>
        <p><font size=3 color=#2F4F4F>is not in DBMS</font></p>
    <%
    }
    }catch( Exception e ){
        e.printStackTrace();
    }
    %>
</td></tr>
<tr><td height=20></td></tr>
<tr><td width="100%" bgcolor=#D3D3D3 align=left>
    <p><font color=#2F4F4F size=2>
        <a href="index.jsp">Edit More Records</a>
    </font></p></td></tr>
</table>
</div>
</body>
</html>

```

## Phonenumber.jsp

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<%% page language="java" import="common.*, database.*, java.io.*, java.util.*"
session="true" %>
<%% include file="../../safe2.jsp" %>

<html>
    <head>
        <title>Telephone Company One--TDWS Master Project CSUSB</title>
        <meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
        <script type="text/javascript" language="javascript"
src="../../common/editphonenumber.js">
        </script>
    </head>
    <body topMargin=0>
        <jsp:useBean id="error" scope="request" class="java.util.Vector"/>
        <jsp:useBean id="editbean" scope="request" class="common.TelcoBean"/>
        <%
            try{
                TelcoBean recordbean = ( common.TelcoBean )session.getAttribute( "recordbean"
);

                String fname = recordbean.getFirstName();
                String mname = recordbean.getMiddleName();
                String lname = recordbean.getLastName();
                String street = recordbean.getStreetName();
                String city = recordbean.getCity();
                String state = recordbean.getState();
                String zipcode = recordbean.getZipCode();
                String phone = recordbean.getPhoneNumber();

            %>

            <%% include file="../../common/menu/menubar.jsp" %>
            <tr><td align=center height=20 width="100%">
                <p><font color=#2F4F4F>Edit Telephone Number</font></p>
            </td></tr>
            <tr><td width="100%" height=20></td></tr>
            <tr>
                <td bgColor=#D3D3D3 class=font3 colSpan=1 width="100%">
                    <p><font size=2 color=#2F4F4F><b>The Current Record</b></font></p>
                </td></tr>
            <tr><td height=20></td></tr>
            <tr><td>
                <%
                    UpdatePhone upd = ( .database.UpdatePhone )session.getAttribute( "upd"
);

                    if( upd.getNo() )
                    {
                        %>
                    </td></tr>
            <tr><td height=10></td></tr>

```



```

<tr><td align=left>
  <div align=center>
    <table border=0 cellpadding=0 cellspacing=0 width="100%">
      <%
        out.println( upd.current() );
      %>
    </table></div>
  </td></tr>
<tr><td width="100%" height=20></td></tr>
<tr>
  <td bgcolor=#D3D3D3 class=font3 colSpan=1 width="100%">
    <p><font size=2 color=#2F4F4F><b>'s <%=phone%> to be edited
with</b></font></p>
  </td></tr>
<tr><td width="100%" height=10></td></tr>
<tr>
  <td align=left width="100%">
    <p><font size=1 color=#8B0000>( *All fields are required )
  </font></p></td></tr>
<tr><td width="100%" height=20></td></tr>
<tr>
  <td>
    <ul>
      <font color="red" size=2>
        <%
          if( !error.isEmpty() )
          {
            for( int i = 0; i < error.size(); i ++ )
            {
              <li><%=error.get( i ) %>
            }
            error.removeAllElements();
          }
        %>
      </font>
    </ul>
  </td>
</tr>
<tr>
  <form name=logonForm method=post action="confirmphonenumber.jsp"
onSubmit="return evaldate()">
    <td align=left>
      <div align=center>
        <table border=0 cellpadding=0 cellspacing=0 width="100%">
          <%@ include file="../../../common/editphone.jsp" %>
          <tr><td height=20></td></tr>
          <tr>
            <td align=left>
              <input type=submit value=Save>
            </td>
          </tr>
        </table></div>
      </td>
    </form>
  </tr>
<tr><td height=10></td></tr>
<tr>
  <form action="index.jsp">
    <td align=center>
      <div align=center>
        <table border=0 cellpadding=0 cellspacing=0 width="100%">
          <tr>
            <td align=left>
              <input type=submit value=Cancel>
            </td>
          </tr>
        </table></div>
      </td>
    </form>
  </tr>
<tr><td align=left>
  <%
    }else{%>
      <div align=center>
        <table border=0 cellpadding=0 cellspacing=0 width="100%">
          <%

```

```

        out.println( upd.current1() );
        %>
    </table></div>
    <font size=2 color=#2F4F4F><b>Telephone Number:</b></font>
    <font size=2 color=#2F4F4F> <%=phone%></font><br>
    <p><font size=3 color=#2F4F4F>is not in DEMS</font></p>
    <% }
    }catch( Exception e ){
        e.printStackTrace();
    }
    %>
</td></tr>
<tr><td height=30></td></tr>
<tr><td width="100%" bgcolor=#D3D3D3 align=left>
    <p><font color=#2F4F4F size=2>
        <a href="index.jsp">Edit More Records</a>
    </font></p></td></tr>
</table>
</div>
</body>
</html>

```

## Confirmphonenumber.jsp

```

<%@ page language="java" import="common.*, database.*, java.util.*" session="true" %>
<%@ include file="../../safe2.jsp" %>

<html>
<head>
    <title>Telephone Company One--TDWS Master Project CSUSB</title>
    <meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
</head>
<body>
    <%@ include file="../../common/variable.jsp" %>
    <%@ include file="../../common/middlecode.jsp" %>
    <%@ include file="../../common/lastcode.jsp" %>

    <jsp:useBean id="error" scope="request" class="java.util.Vector"/>
    <jsp:useBean id="editbean" scope="request" class="common.TelcoBean"/>
    <jsp:setProperty name="editbean" property="middlecode" value="<%= middleCode%>"/>
    <jsp:setProperty name="editbean" property="lastcode" value="<%= lastCode%>"/>
    <jsp:setProperty name="editbean" property="phoneNumber" value="<%=
phoneNumber%>"/>
    <%
        try{
            TelcoBean recordbean = ( common.TelcoBean )session.getAttribute( "recordbean"
);

            String firstname = recordbean.getFirstName();
            String middlename = recordbean.getMiddleName();
            String lastname = recordbean.getLastName();
            String streetname = recordbean.getStreetName();
            String City = recordbean.getCity();
            String State = recordbean.getState();
            String myzip = recordbean.getZipCode();
            String phone = recordbean.getPhoneNumber();

        %>

    <%@ include file="../../common/menu/menubar.jsp" %>
    <tr><td width="100%" bgcolor=#D3D3D3 align=left>
        <p><font color=#2F4F4F size=2><b>Update Record</b></font></p></td></tr>
    <tr><td height=20></td></tr>
    <tr><td align=left>
        <%
            UpdatePhone upd = ( database.UpdatePhone )session.getAttribute( "upd" );
            upd = new UpdatePhone( recordbean, editbean );
            session.setAttribute( "upd", upd );

            if( upd.getNo() )
            {
                if( mc && lc && !(phone.trim()).equals( phoneNumber.trim() ) )
                {
                    if( !upd.getEdityes() )

```

```

    {
        %>
        </td></tr>
        <tr><td align=left height=20>
            <p><font size=3 color=#2F4F4F>The current record</font></p>
        </td></tr>
        <tr><td height=10></td></tr>
        <tr><td align=left>
            <div align=center>
                <table border=0 cellPadding=0 cellSpacing=0 width="100%">
                    <%
                        out.println( upd.current() );
                    %>
                </table></div>
            </td></tr>
            <tr><td width="100%" height=20></td></tr>
            <tr>
                <td bgColor=#D3D3D3 class=font3 colSpan=1 width="100%">
                    <p><font size=2 color=#2F4F4F><b>'s <%=phone%> to be edited
with</b></font></p>
                </td></tr>
                <tr><td width="100%" height=20></td></tr>
                <tr>
                    <form action="editphonenumber.jsp">
                        <td align=center>
                            <div align=center>
                                <table border=0 cellPadding=0 cellSpacing=0 width="100%">
                                    <tr><td align=left height=20 width="100%">
                                        <p><font size=3 color=#2F4F4F>The New Telephone
Number</font></p>
                                    <tr>
                                        <td align=left>
                                            <font size=2 color=#2F4F4F><b>Telephone Number: </b></font>
                                            <font size=2 color=#2F4F4F><%=phoneNumber%></font><br>
                                        </td></tr>
                                        <tr>
                                            <td align=center>
                                                <div align=center>
                                                    <table border=0 cellPadding=0 cellSpacing=0
width="100%">
                                                        <tr>
                                                            <td align=left>
                                                                <input type=submit value=OK>
                                                            </td>
                                                            <td align=right>
                                                                <input type=submit value=Cancel>
                                                            </td>
                                                        </tr>
                                                    </table></div>
                                                </td>
                                            </tr>
                                        </table></div>
                                    </td></tr>
                                </table>
                            </div>
                        </td></tr>
                    </form>
                </tr>
                <tr><td height=10></td></tr>
                <tr>
                    <form action="index.jsp">
                        <td align=center>
                            <div align=center>
                                <table border=0 cellPadding=0 cellSpacing=0 width="100%">
                                    <tr>
                                        <td align=left>
                                            <input type=submit value=Cancel>
                                        </td>
                                        <td align=right>
                                            <input type=submit value=OK>
                                        </td>
                                    </tr>
                                </table></div>
                            </td>
                        </tr>
                    </form>
                </tr>
                <tr><td align=left>
                    <%
                        }else{%>
                            <p><font size=3 color=#2F4F4F>The Telephone Number </font>
                                <font size=2 color=#2F4F4F><b><%=phoneNumber%></b> </font>
                                <font size=3 color=#2F4F4F>is being using by </font></p>
                            <div align=center>
                                <table border=0 cellPadding=0 cellSpacing=0 width="100%">
                                    <%
                                        out.println( upd.phonenumberused() );
                                    %>
                                </table>
                            </div>
                        </td>
                    </tr>
                </tr>
            </td>
        </tr>
    }
}

```

```

        </table></div>
    <%    }
    }else{
        if( !mc || !lc )
        {
            error.addElement( "Invalid Telephone Number" );
        }
        if( (phone.trim()).equals( phoneNumber.trim() ) )
        {
            error.addElement( "The new phone number is the same as the
current phone number" );
        }
    }
    if( error.size() > 0 )
    {
        <jsp:forward page="phonenumbers.jsp"/>
    }
}
else{
    <p><font size=3 color=#2F4F4F>The record </font></p>
    <div align=center>
        <table border=0 cellpadding=0 cellspacing=0 width="100%">
            <%
                out.println( upd.current1() );
            %>
        </table></div>
        <font size=2 color=#2F4F4F><b>Telephone Number:</b></font>
        <font size=2 color=#2F4F4F> <%=phone%></font><br>
        <p><font size=3 color=#2F4F4F>is not in DBMS </font></p>
    <%    }
    }catch( Exception e ){
        e.printStackTrace();
    }
    %>
    </td></tr>
    <tr><td height=30></td></tr>
    <tr><td width="100%" bgcolor=#D3D3D3 align=left>
        <p><font color=#2F4F4F size=2>
            <a href="index.jsp">Edit More Records</a>
        </font></p></td></tr>
    </table>
</div>
</body>
</html>

```

## Editphonenumbers.jsp

```

<%@ page language="java" import="common.*, database.*, java.util.*" session="true" %>
<%@ include file="../../../safe2.jsp" %>

<html>
<head>
    <title>Telephone Company One--TDWS Master Project CSUSB</title>
    <meta content="text/html; charset=iso-8895-1" http-equiv=Content-Type>
</head>
<body>
    <%
        try{
            TelcoBean recordbean = ( common.TelcoBean )session.getAttribute( "recordbean"
);

            String firstname = recordbean.getFirstName();
            String middlename = recordbean.getMiddleName();
            String lastname = recordbean.getLastName();
            String streetname = recordbean.getStreetName();
            String City = recordbean.getCity();
            String State = recordbean.getState();
            String myzip = recordbean.getZipCode();
            String phone = recordbean.getPhoneNumber();

        }
        %>
        <%@ include file="../../../common/menu/menubar.jsp" %>
        <tr><td width="100%" bgcolor=#D3D3D3 align=left>
            <p><font color=#2F4F4F size=2><b>Update Record</b></font></p></td></tr>

```

```

<tr><td height=20></td></tr>
<tr><td align=left>
<%
    UpdatePhone upd = ( database.UpdatePhone )session.getAttribute( "upd" );

    if( upd.getNo() )
    {
        if( !upd.getEdityes() )
        {
            %>
</td></tr>
<tr><td align=left height=20>
    <p><font size=3 color=#2F4F4F>The old record</font></p>
</td></tr>
<tr><td height=10></td></tr>
<tr><td align=left>
    <div align=center>
        <table border=0 cellPadding=0 cellSpacing=0 width="100%">
            <%
                out.println( upd.current() );
            %>
        </table></div>
</td></tr>
<tr><td width="100%" height=20></td></tr>
<tr>
    <td bgcolor=#D3D3D3 class=font3 colSpan=1 width="100%">
        <p><font size=2 color=#2F4F4F><b>'s <%=phone%> has already
edited</b></font></p>
    </td></tr>
<tr><td width="100%" height=20></td></tr>
<tr><td align=left>
    <%
        upd.execute_editPhone();
        out.println( upd.getMessage() );
    %>
</td></tr>
<tr><td align=left>
    <%
        }else{%>
            <p><font size=3 color=#2F4F4F>The Telephone Number </font>
            <font size=2
color=#2F4F4F><b><%=upd.getEditbean().getPhoneNumber()%></b> </font>
            <font size=3 color=#2F4F4F>is being using by </font></p>
            <div align=center>
                <table border=0 cellPadding=0 cellSpacing=0 width="100%">
                    <%
                        out.println( upd.phonenumberused() );
                    %>
                </table></div>
            <%
        }
    }else{%>
        <p><font size=3 color=#2F4F4F>The record </font></p>
        <div align=center>
            <table border=0 cellPadding=0 cellSpacing=0 width="100%">
                <%
                    out.println( upd.current1() );
                %>
            </table></div>
            <font size=2 color=#2F4F4F><b>Telephone Number:</b></font>
            <font size=2 color=#2F4F4F> <%=phone%></font><br>
            <p><font size=3 color=#2F4F4F>is not in DBMS</font></p>
        <%
    }
}catch( Exception e ){
    e.printStackTrace();
}
%>
</td></tr>
<tr><td height=30></td></tr>
<tr><td width="100%" bgcolor=#D3D3D3 align=left>
    <p><font color=#2F4F4F size=2>
        <a href="index.jsp">Edit More Records</a>
    </font></p></td></tr>
</table>
</div>
</body>
</html>

```

## Telephone Company Two Package:

### Build.xml

```
<!DOCTYPE project [
  <!ENTITY commonTargets SYSTEM "../common/targets.xml">
  <!ENTITY managertargets SYSTEM "../common/managertargets.inc">
]>

<project name="JAX-RPC Telephone Directory Web Service"
  default="build"
  basedir=". ">
  <description>
    JAX-RPC Telephone Directories Web Services Build File
  </description>

  <property file="../common/build.properties"/>
  <property file="${USER_HOME}/build.properties"/>
  <property file="build.properties"/>

  <property name="compile_debug" value="true"/>
  <property name="compile_deprecation" value="false"/>
  <property name="compile_optimize" value="true"/>

  <path id="compile.classpath">
    <!-- Include all JAR files that will be included in /WEB-INF/lib -->
    <!-- Include all elements that Tomcat exposes to applications -->
    <pathelement location="${CATALINA_HOME}/common/classes"/>
    <fileset dir="${CATALINA_HOME}/common/endorsed">
      <include name="*.jar"/>
    </fileset>
    <fileset dir="${CATALINA_HOME}/common/lib">
      <include name="*.jar"/>
    </fileset>
    <pathelement location="${CATALINA_HOME}/shared/classes"/>
    <fileset dir="${CATALINA_HOME}/shared/lib">
      <include name="*.jar"/>
    </fileset>
  </path>

  &managertargets; <!-- The ant targets are in ../common/managertargets.xml -->
  &commonTargets; <!-- The ant targets are in ../common/targets.xml -->

  <target name="build" depends="all"
    description="Executes the targets needed to build the JAX-RPC Web Service.">
    <echo message="Executes the targets needed to build the JAX-RPC Web Service."/>
  </target>
</project>
```

### target.xml

```
<!-- Initialization target -->
<target name="init" description="The initialization target">
  <echo message="Creating the time stamp..."/>
  <tstamp/>
</target>

<!-- Clean Target -->
<target name="clean" depends="init"
  description="Removes the build and dist directories.">
  <echo message="Deleting old build and dist directories..."/>
  <delete dir="${build}"/>
  <delete dir="${dist}"/>
</target>

<!-- Prepare Target -->
<target name="prepare" depends="clean"
  description="Creates the build and dist directories" >
  <echo message="Creating the required directories...." />
  <mkdir dir="${build}/client" />
  <mkdir dir="${build}/server" />
```

```

    <mkdir dir="${build}/shared" />
    <mkdir dir="${build}/wsdeploy-generated" />
    <mkdir dir="${dist}" />
    <mkdir dir="${build}/WEB-INF/classes" />
    <mkdir dir="${build}/WEB-INF/lib" />

    <!-- Copy web.xml over from web/WEB-INF -->
    <copy todir="${build}/WEB-INF/lib">
      <fileset dir="web/WEB-INF/lib">
        <include name="**/*.jar"/>
      </fileset>
    </copy>
  </target>

<!--Set-we-scripts Target -->
  <target name="set-ws-scripts"
    description="Sets the value of the wscompile and wsdeploy properties for this build
file" >
    <echo message="Sets the value of the wscompile and wsdeploy properties..." />
    <condition property="script-suffix" value="bat">
      <os family="windows"/>
    </condition>
    <condition property="script-suffix" value="sh">
      <not>
        <os family="windows"/>
      </not>
    </condition>
    <property name="wscompile" value="${wscompile-path}/wscompile.${script-suffix}" />
    <property name="wsdeploy" value="${wscompile-path}/wsdeploy.${script-suffix}" />
  </target>

<!-- Compile Target -->
  <target name="compile-server" depends="prepare"
    description="Compiles the server-side java source code">
    <echo message="Compiling the server-side java source code...." />

    <!-- Copy application resources -->
    <copy todir="${build}/WEB-INF/classes">
      <fileset dir="${src}" excludes="**/*.java">
        <include name="**/*.*" />
      </fileset>
    </copy>

    <!-- Compile Java classes as necessary -->
    <javac srcdir="${src}"
      destdir="${build}/shared"
      includes="**/*.java"
      excludes="**/*Fetcher.java"
      debug="${compile_debug}"
      deprecation="${compile_deprecation}"
      optimize="${compile_optimize}"
      fork="yes">
      <classpath refid="compile.classpath" />
      <classpath path="${common}" />
    </javac>
  </target>

<!-- Copy Common Target-->
  <target name="copy-common"
    description="Copies common class files to jaxrpc web service">
    <echo message="Copies /common dir classes..." />
    <copy todir="${build}/shared">
      <fileset dir="${common}" />
    </copy>
  </target>

<!-- Target setup-web-inf -->
  <target name="setup-web-inf" depends="compile-server, copy-common"
    description="Copies files to build/WEB-INF">
    <echo message="Setting up ${build}/WEB-INF...." />
    <copy todir="${build}/WEB-INF/classes">
      <fileset dir="${build}/shared/" />
      <fileset dir="${build}/server/" />
    </copy>
    <copy todir="${build}">
      <fileset dir="${web}">
        <include name="**/*.*" />
      </fileset>
    </copy>
  </target>

```

```

        </fileset>
    </copy>
    <copy file="web/web.xml" todir="${build}/WEB-INF" />
    <copy file="jaxrpc-ri.xml" todir="${build}/WEB-INF" />
</target>

<!-- Target Package -->
<target name="package" depends="setup-web-inf"
    description="Packages the WAR file">
    <echo message="Packaging the WAR..." />
    <jar jarfile="dist/${portable-war}" >
        <fileset dir="${build}" includes="WEB-INF/**" />
    </jar>
</target>

<!-- Target process-war -->
<target name="process-war" depends="set-ws-scripts, package"
    description="Runs wsdeploy to generate the ties and create a deployable WAR file">
    <echo message="Running wsdeploy to build JAXRPC Web Service on server side..." />
    <exec executable="${wsdeploy}">
        <arg line="-tmpdir"/>
        <arg line="${build}/wsdeploy-generated"/>
        <arg line="-o"/>
        <arg line="${dist}/${deployable-war}"/>
        <arg line="${dist}/${portable-war}"/>
        <arg line="-verbose"/>
        <arg line="-keep"/>
    </exec>
</target>

<!-- Target generate-stubs -->
<target name="generate-stubs" depends="set-ws-scripts"
    description="Runs wscompile to generate the client stub classes">
    <echo message="Running wscompile to build local JAXRPC Web Service on client
side..." />
    <exec executable="${wscompile}">
        <arg line="-gen:client"/>
        <arg line="-d ${build}/client"/>
        <arg line="-classpath ${build}/shared"/>
        <arg line="-g"/>
        <arg line="-keep"/>
        <arg line="config.xml"/>
    </exec>
</target>

<!-- Compile-client Target -->
<target name="compile-client" depends="generate-stubs"
    description="Compiles the client-side source code" >
    <echo message="Compiling the client source code..." />
    <javac srcdir="${src}"
        destdir="${build}/client"
        includes="**/*Fetcher.java"
        debug="${compile_debug}"
        deprecation="${compile_deprecation}"
        optimize="${compile_optimize}"
        fork="yes">
        <classpath path="${jwsdp-jars}:build/shared:build/client:${common}"/>
    </javac>
</target>

<!-- Jar-client Target -->
<target name="jar-client" depends="compile-client"
    description="Builds the JAR file that contains the client">
    <echo message="Building the JAXRPC client JAR file..." />
    <jar jarfile="${dist}/${client-jar}" >
        <fileset dir="${build}/client" />
        <fileset dir="${build}/shared" >
            <exclude name="**/*Impl*" />
        </fileset>
    </jar>
</target>

<!-- Target run -->
<target name="run"
    description="Tests the JAXRPC client">
    <echo message="Testing the JAXRPC Client program..." />

```



```

        <java fork="on"
            classpath="${dist}/${client-jar}:${jwsdp-jars}"
            classname="${client_class}" >
        </java>
    </target>

<!-- Target all -->
    <target name="all" depends="undeploy,
        deploy,
        jar-client,
        debug"/>

<!-- Target Debug -->
    <target name="debug" depends="set-ws-scripts"
        description="Displays values of some of the properties of this build file">
        <echo message=" CATALINA_HOME = ${CATALINA_HOME}" />
        <echo message=" JWSDP_HOME = ${JWSDP_HOME}" />
        <echo message=" USER_HOME = ${USER_HOME}" />
        <echo message=" username = ${username}" />
        <echo message=" password = ${password}" />
        <echo message=" url = ${url}" />
        <echo message=" wscompile = ${wscompile}" />
        <echo message=" wsdeploy = ${wsdeploy}" />
        <echo message=" webservice = ${webservice}" />
        <echo message=" context_path = ${context_path}" />
        <echo message=" war_path = ${war_path}" />
        <echo message=" build = ${build}" />
        <echo message=" scr = ${src}" />
        <echo message=" dist = ${dist}" />
        <echo message=" client_class = ${client_class}" />
        <echo message=" client-jar = ${client-jar}" />
        <echo message=" portable-war = ${portable-war}" />
        <echo message=" deployable-war = ${deployable-war}" />
        <echo message=" common = ${common}" />
        <echo message=" war_path = ${war_path}" />
        <echo message=" jwsdp-jars = ${jwsdp-jars}" />
    </target>

```

## JAXRPCTelcoImpl.java

```

// Master Project: Telephone Directories Web Services
// JAXRPC Telephone Directory Web Service
// Adviser: Dr. David Turner
// Programmer: Hua Sun
// Date: March 30, 2003
//
// jaxrpc2.JAXRPCTelcoImpl.java

package jaxrpc2;

import java.io.*;
import common.*;

/**
 * Implementation of Interface
 */
public class JAXRPCTelcoImpl implements JAXRPCTelcoIF
{
    public DBHandler dbHandler;
    public int size = 0;
    public PhoneNumberItems[] items;

    public PhoneNumberItems[] getPhoneNumber( String fn,
        String mn,
        String ln,
        String city )
    {
        dbHandler = new DBHandler( fn, mn, ln, city );
        size = dbHandler.getSize();
        items = new PhoneNumberItems[ size ];
        items = dbHandler.handler();
    }
}

```

```

        return items;
    }

    public int getSize( String fn,
                        String mn,
                        String ln,
                        String city )
    {
        dbHandler = new DBHandler( fn, mn, ln, city );
        size = dbHandler.getSize();

        return size;
    }
}

```

## JAXRPCTelcoFetcher.java

```

// Master Project: Telephone Directories Web Services
// JAXRPC Telephone Directory Web Service
// Adviser: Dr. David Turner
// Programmer: Hua Sun
// Date: March 30, 2003
//
// jaxrpc2.JAXRPCTelcoFetcher.java

package jaxrpc2;

import java.io.*;
import common.*;

/**
 * Used by the information aggregator to make a method call
 */
public class JAXRPCTelcoFetcher
{
    public static void main( String[] args )
    {
        try{
            System.out.println( "*** getPhoneNumber( \"Hua\", \"\", \"Sun\", \"city2\" )=" +
                                getPhoneNumber( "Hua", "", "Sun", "city2" ) );
            System.out.println( "*** getSize( \"Hua\", \"\", \"Sun\", \"city2\" )=" +
                                getSize( "Hua", "", "Sun", "city2" ) );
        }catch( Exception ex ) {
            System.out.println( "*** HelloClient main()'s ex=" + ex );
            ex.printStackTrace();
        }
    }

    public static PhoneNumberItems[] getPhoneNumber( String fn, String mn, String ln,
String city )
    {
        PhoneNumberItems[] items = null;
        int size = 0;

        try {
            JAXRPCTelcoIF_Stub stub = (JAXRPCTelcoIF_Stub)(new
JAXRPCTelco_Impl().getJAXRPCTelcoIFPort());
            size = stub.getSize( fn, mn, ln, city );
            System.out.println( "*** HelloClient size=" + size );
            items = new PhoneNumberItems[ size ];
            System.out.println( "*** HelloClient items1=" + items );
            items = stub.getPhoneNumber( fn, mn, ln, city );

            int i = 0;
            while( i < size )
            {
                System.out.println( "*** HelloClient getPhoneNumber() items[ i ].getFname()="
+
                                items[i].getFname() );

                i ++;
            }
        }
    }
}

```

```

        System.out.println( "*** HelloClient items2=" + items );
    } catch (Exception ex) {
        System.out.println( "*** HelloClient getPhoneNumber()'s ex=" + ex );
        ex.printStackTrace();
    }

    System.out.println( "*** HelloClient items3=" + items );

    return items;
}

public static int getSize( String fn, String mn, String ln, String city )
{
    int size = 0;

    try {
        JAXRPCTelcoIF_Stub stub = (JAXRPCTelcoIF_Stub)(new
JAXRPCTelco_Impl().getJAXRPCTelcoIFPort());
        size = stub.getSize( fn, mn, ln, city );
        System.out.println( "*** HelloClient size=" + size );
    } catch (Exception ex) {
        System.out.println( "*** HelloClient getSize()'s ex=" + ex );
        ex.printStackTrace();
    }

    return size;
}
}

```

## JAXRPCTelcoIF.java

```

// Master Project: Telephone Directories Web Services
// JAXRPC Telephone Directory Web Service
// Adviser: Dr. David Turner
// Programmer: Hua Sun
// Date: March 30, 2003
//
// jaxrpc2.JAXRPCTelcoIF.java

package jaxrpc2;

import java.rmi.*;
import common.*;

/**
 * Declare Precedure's name
 */
public interface JAXRPCTelcoIF extends Remote
{
    public PhoneNumberItems[] getPhoneNumber( String fn,
                                                String mn,
                                                String ln,
                                                String city ) throws RemoteException;

    public int getSize( String fn,
                        String mn,
                        String ln,
                        String city ) throws RemoteException;
}

```

## Database Package:

### CreateTable.java

```
// Date: March 11, 2003
package database;
import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.*;
import common.*;
/**
 * Create person table and phone table
 */
public class CreateTable
{
    protected String query;
    protected int count;
    protected int size;
    protected int length;
    protected String lineEnd;
    protected StringTokenizer token;
    protected Statement statement;
    protected ResultSet resultSet;
    protected ResultSetMetaData rsmd;
    protected Connection connection;
    protected DBConnection dbConnection;

    public CreateTable()
    {
        this.query = "";
        this.count = 0;
        this.size = 0;
        this.length = 0;
        this.lineEnd = System.getProperty( "line.separator" );
        this.statement = null;
        this.resultSet = null;
        this.rsmd = null;
        this.connection = null;
        this.dbConnection = new DBConnection( "telcol" );
    }

    public void getConnnToDatabase()
    {
        try{
            connection = dbConnection.getConnection();
            statement = connection.createStatement();
        }catch( SQLException se){
            se.printStackTrace();
        }catch( Exception exc ){
            exc.printStackTrace();
        }
    }

    public void create_person()
    {
        query = "CREATE TABLE person(" + lineEnd +
            "    FirstName varchar(80) NOT NULL," + lineEnd +
            "    MiddleName varchar(80)," + lineEnd +
            "    LastName varchar(80) NOT NULL," + lineEnd +
            "    Street varchar(80) NOT NULL," + lineEnd +
            "    City varchar(80) NOT NULL," + lineEnd +
            "    State varchar(80) NOT NULL," + lineEnd +
            "    ZipCode varchar(80) NOT NULL," + lineEnd +
            "    Account_id serial4 PRIMARY KEY" + lineEnd +
            ");" + lineEnd;

        try{
            statement.executeUpdate( query );
        }catch( SQLException ex ){
            ex.printStackTrace();
        }catch( Exception exc ){
            exc.printStackTrace();
        }
    }
}
```

```

    }

    public void create_phone()
    {
        query = "CREATE TABLE phone(" + lineEnd +
            "    person_id int4 REFERENCES person, " + lineEnd +
            "    phoneNumber varchar(80) PRIMARY KEY" + lineEnd +
            ");" + lineEnd;

        try{
            statement.executeUpdate( query );
        }catch( SQLException ex ){
            ex.printStackTrace();
        }catch( Exception exc ){
            exc.printStackTrace();
        }
    }

    public void closeConnToDatabase()
    {
        try{
            resultSet.close();
            statement.close();
        }catch( SQLException se ){
            se.printStackTrace();
        }catch( NullPointerException ne ){
            ne.printStackTrace();
        }
        dbConnection.close();
    }

    public void execute_creation()
    {
        getConnToDatabase();
        create_person();
        create_phone();
        closeConnToDatabase();
    }
}

```

## Review.java

```

// Date: March 11, 2003

package database;

import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.*;
import common.*;

/**
 * Review records stored in database
 */
public class Review
{
    protected String query;
    protected String temp;
    protected String temp1;
    protected String message;
    protected boolean yes;
    protected boolean no;
    protected boolean yesno;
    protected int count;
    protected int size;
    protected int length;
    protected String lineEnd;

    protected Statement statement;
    protected ResultSet resultSet;
    protected ResultSetMetaData rsmd;
    protected Connection connection;
    protected DBConnection dbConnection;
}

```

```

public Review()
{
    this.query = "";
    this.temp = "";
    this.temp1 = "";
    this.message = "";
    this.yes = false;
    this.no = false;
    this.yesno = false;
    this.count = 0;
    this.size = 0;
    this.length = 0;
    this.lineEnd = System.getProperty( "line.separator" );

    this.statement = null;
    this.resultSet = null;
    this.rsmd = null;
    this.connection = null;
    this.dbConnection = new DBConnection( "telcol" );
}

public void getConnToDatabase()
{
    try{
        connection = dbConnection.getConnection();
        statement = connection.createStatement();
    }catch( SQLException se){
        se.printStackTrace();
    }catch( Exception exc ){
        exc.printStackTrace();
    }
}

public void review_person()
{
    String str = "";

    query = "SELECT * from person " + lineEnd +
            "ORDER BY account_id;" + lineEnd;

    try{
        no = statement.execute( query );

        if( no )
        {
            message = "";

            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            count = rsmd.getColumnCount();

            for( int i = 1; i <= count; i ++ )
            {
                message = message +
                    "<th><p><font color=#2F4F4F size=2>" + rsmd.getColumnName( i ) +
                    "</p></font></th>";
            }
            message = "<tr>" + message + "</tr>";

            while( resultSet.next() )
            {
                str = str + "<tr>";
                String str1 = "";

                for( int i = 1; i <= count; i ++ )
                {
                    if( !resultSet.getString( i ).equals( "" ) )
                    {
                        str1 = str1 +
                            "<td><p><font color=#2F4F4F size=2>" + resultSet.getString( i
) +
                            "</p></font></td>";
                    }
                    else{
                        str1 = str1 +
                            "<td>&nbsp;" +
                            "</td>";
                    }
                }
            }
        }
    }
}

```

```

        }
        str = str + str1 + "</tr>";
    }

    message = message + str;
}
}catch( SQLException se ){
    se.printStackTrace();
}
}

public void review_phone()
{
    String str = "";

    query = "SELECT * from phone " + lineEnd +
            "ORDER BY person_id;" + lineEnd;

    try{
        no = statement.execute( query );

        if( no )
        {
            message = "";
            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            count = rsmd.getColumnCount();

            for( int i = 1; i <= count; i ++ )
            {
                message = message +
                    "<th><p><font color=#2F4F4F size=2>" + rsmd.getColumnName( i ) +
                    "</p></font></th>";
            }
            message = "<tr>" + message + "</tr>";

            while( resultSet.next() )
            {
                str = str + "<tr>";
                String str1 = "";

                for( int i = 1; i <= count; i ++ )
                {
                    str1 = str1 +
                        "<td><p><font color=#2F4F4F size=2>" + resultSet.getString( i ) +
                        "</p></font></td>";
                }
                str = str + str1 + "</tr>";
            }

            message = message + str;
        }
    }catch( SQLException se ){
        se.printStackTrace();
    }
}

public String getMessage()
{
    return message;
}

public void closeConnToDatabase()
{
    try{
        resultSet.close();
        statement.close();
    }catch( SQLException se ){
        se.printStackTrace();
    }catch( NullPointerException ne ){
        ne.printStackTrace();
    }

    dbConnection.close();
}

```

```

public void execute_reviewC()
{
    getConnToDatabase();
    review_person();
    closeConnToDatabase();
}

public void execute_reviewP()
{
    getConnToDatabase();
    review_phone();
    closeConnToDatabase();
}
}

```

## DBConnection.java

```

// Date: March 10, 2003

package database;

import java.sql.*;
import java.io.*;
import java.util.*;

/**
 * Connecte to database server
 */
public class DBConnection
{
    protected String connURL;
    protected String user;
    protected String password;
    protected String port;
    protected String database;
    protected Connection connection;
    protected String propertiesFileName;
    protected FileInputStream fis;
    protected File file;
    protected Properties properties;

    public DBConnection()
    {
        this.properties = new Properties();
        this.port = "5432";
        this.connURL = "";
        this.user = "hsun";
        this.password = "";
        this.database = "";
        this.connection = null;
    }

    public DBConnection( String database )
    {
        this.properties = new Properties();
        this.port = "5432";
        this.connURL = "";
        this.user = "hsun";
        this.password = "";
        this.database = database;
        this.connection = null;
    }

    public Connection getConnection()
    {
        if( connection == null )
        {
            try{
                Class.forName( "org.postgresql.Driver" );
                connURL = "jdbc:postgresql://127.0.0.1:" + port + "/" + database;
                connection = DriverManager.getConnection( connURL,

```



```

        user,
        password );

    if( connection == null )
    {
        System.out.println( connURL );
        System.out.println( "connection is null" );
    }

    System.out.println( "Go" );
} catch( Exception exc ){
    exc.printStackTrace();
}
}
else{
    connection = null;

    try{
        Class.forName( "org.postgresql.Driver" );
        connURL = "jdbc:postgresql://127.0.0.1:" + port + "/" + database;

        connection = DriverManager.getConnection( connURL,
                                                    user,
                                                    password );

        System.out.println( "Go" );
    } catch( Exception exc ){
        exc.printStackTrace();
    }
}

return connection;
}

public void close()
{
    try{
        if( connection != null )
        {
            connection.close();
            System.out.println( "*** DBConnection close() connection is closed." );
        }
        else{
            System.out.println( "*** DBConnection close() connection is null." );
        }
    }catch( SQLException se ){
        se.printStackTrace();
    }catch( NullPointerException ne ){
        ne.printStackTrace();
    }
}
}
}

```

## Deletion.java

```

// Date: March 11, 2003

package database;

import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.*;
import common.*;

/**
 * Delete customer from database
 */
public class Deletion
{
    protected String query;
    protected String query1;
    protected String temp;
    protected String temp1;
    protected String message;
}

```

```

protected boolean yes;
protected boolean no;
protected boolean yesno;
protected int count;
protected int size;
protected int length;
protected String lineEnd;

protected StringTokenizer token;
protected Statement statement;
protected ResultSet resultSet;
protected ResultSetMetaData rsmd;
protected Connection connection;
protected DBConnection dbConnection;
protected TelcoBean telcobean;

public Deletion()
{
    this.query = "";
    this.query1 = "";
    this.temp = "";
    this.temp1 = "";
    this.message = "";
    this.yes = false;
    this.no = false;
    this.yesno = false;
    this.count = 0;
    this.size = 0;
    this.length = 0;
    this.lineEnd = System.getProperty( "line.separator" );

    this.statement = null;
    this.resultSet = null;
    this.rsmd = null;
    this.connection = null;
    this.dbConnection = new DBConnection( "telco1" );
}

public Deletion( TelcoBean bean )
{
    this.query = "";
    this.query1 = "";
    this.temp = "";
    this.temp1 = "";
    this.message = "";
    this.yes = false;
    this.no = false;
    this.yesno = false;
    this.count = 0;
    this.size = 0;
    this.length = 0;
    this.lineEnd = System.getProperty( "line.separator" );

    this.statement = null;
    this.resultSet = null;
    this.rsmd = null;
    this.connection = null;
    this.telcobean = bean;
    this.dbConnection = new DBConnection( "telco1" );
}

public void getConnToDatabase()
{
    try{
        connection = dbConnection.getConnection();
        statement = connection.createStatement();
    }catch( SQLException se){
        se.printStackTrace();
    }catch( Exception exc ){
        exc.printStackTrace();
    }
}

public void getCity()
{
    temp = telcobean.getCity().trim();
}

```

```

token = new StringTokenizer( temp, "," );

try{
    String city = token.nextToken().trim();
    telcobean.setCity( city );

    String state = token.nextToken().trim();
    telcobean.setState( state );
}catch( NoSuchElementException e ){
    e.printStackTrace();
}
}

public void getAccoutID()
{
    getConnToDatabase();
    temp = "1";
    temp1 = "";
    int num = 0;
    int num1 = 1;

    query = "SELECT account_id from person;";
    try{
        yes = statement.execute( query );
        if( yes )
        {
            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            size = rsmd.getColumnCount();
            count = resultSet.getFetchSize();

            if( count > 0 )
            {
                while( resultSet.next() )
                {
                    for( int i = 1; i <=size; i ++ )
                    {
                        num = Integer.parseInt( resultSet.getString( i ) );
                        if( num1 < num )//compareTo( temp1 ) < 0 )
                        {
                            num1 = num;//temp = temp1;
                        }
                    }
                    int id = num1;//Integer.parseInt( temp );
                    id = id + 1;
                    temp = Integer.toString( id );
                    telcobean.setAccount_id( temp );
                }
            }
            else{
                telcobean.setAccount_id( Integer.toString( num1 ) );//"1" );
            }
        }
    }catch( SQLException se ){
        se.printStackTrace();
    }catch( Exception e ){
        e.printStackTrace();
    }
    yes = false;
}

public void getQuery()
{
    try{
        query1 = "SELECT person.firstname, person.middlename, person.lastname,
person.street, person.city, " +
        " person.state, person.zipcode, person.account_id from person " + lineEnd +
        "WHERE person.account_id=phone.person_id" + lineEnd +
        "AND person.firstname LIKE '%" + telcobean.getFirstName().trim() + "%' " +
lineEnd +
        "AND person.middlename LIKE '%" + telcobean.getMiddleName().trim() + "%' " +
lineEnd +
        "AND person.lastname LIKE '%" + telcobean.getLastName().trim() + "%' " + lineEnd
+
        "AND phone.phonenumber LIKE '%" + telcobean.getPhoneNumber().trim() + "%' " +
lineEnd +

```

```

        "ORDER BY person.account_id;" + lineEnd;
    }catch( Exception e ){
        e.printStackTrace();
    }
}

/**      query1 = "SELECT firstname, middlename," +
            " lastname, street, city," +
            " state, zipcode, account_id" +
            " from person" + lineEnd;

    if( (telcobean.getFirstName()).equals( "" ) && (telcobean.getMiddleName()).equals(
"" ) )
    {
        query1 = query1 +
            " WHERE lastname='" + telcobean.getLastName() + "'; " + lineEnd;
    }else if( (telcobean.getFirstName()).equals( "" ) &&
!(telcobean.getMiddleName()).equals( "" ) )
    {
        query1 = query1 +
            " WHERE middlename='" + telcobean.getMiddleName() + "' " + lineEnd
+
            " AND lastname='" + telcobean.getLastName() + "'; " + lineEnd;
    }else
    {
        query1 = query1 +
            " WHERE firstname='" + telcobean.getFirstName() + "' " + lineEnd +
            " AND middlename='" + telcobean.getMiddleName() + "' " + lineEnd +
            " AND lastname='" + telcobean.getLastName() + "'; " + lineEnd;
    }
}

// if customer exists
public void isyes()
{
    getConnToDatabase();
    getQuery();

    try{
        yes = statement.execute( query1 );
        if( yes )
        {
            resultSet = statement.getResultSet();
            count = resultSet.getFetchSize();
            yes = false;

            if( count > 0 )
            {
                yes = true;
            }else{
                yes = false;
            }
        }
    }catch( SQLException se ){
        se.printStackTrace();
    }catch( Exception e ){
        e.printStackTrace();
    }
    closeConnToDatabase();
}

// if customer exists
public void isid()
{
    getConnToDatabase();
    query = "SELECT account_id from person;";

    try{
        no = statement.execute( query );
        if( no )
        {
            resultSet = statement.getResultSet();
            count = resultSet.getFetchSize();
            rsmd = resultSet.getMetaData();
            size = rsmd.getColumnCount();
            no = false;

```

```

        if( count > 0 )
        {
            int i = 0;
            while( resultSet.next() )
            {
                for( i = 1; i <=size; i ++ )
                {
                    if( (resultSet.getString( i ).trim()).equals(
telcobean.getAccount_id().trim()) )
                    {
                        no = true;
                    }
                }
            }
        }else{
            no = false;
        }
    }
} catch( SQLException se ){
    se.printStackTrace();
} catch( Exception e ){
    e.printStackTrace();
}
}
closeConnToDatabase();
}

public void search_person()
{
    getConnToDatabase();
    getQuery();
    String str = "";
    String id = "";

    try{
        no = statement.execute( query1 );

        if( no )
        {
            message = "";

            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            count = rsmd.getColumnCount();

            for( int i = 1; i <= count; i ++ )
            {
                if( i != count )
                {
                    message = message +
                        "<th><p><font color=#2F4F4F size=2>" + rsmd.getColumnName( i ) +
                        "</p></font></th>";
                }
            }

            message = "<tr>" + message + "</tr>";

            while( resultSet.next() )
            {
                if( !(resultSet.getString( count ).trim()).equals( id ) )
                {
                    str = str + "<tr>";
                    String str1 = "";

                    for( int i = 1; i <= count; i ++ )
                    {
                        if( i != count )
                        {
                            if( !resultSet.getString( i ).equals( " " ) )
                            {
                                str1 = str1 +
                                    "<td><p><font color=#2F4F4F size=2>" + resultSet.getString(
i ) +
                                    "</p></font></td>";
                            }else{
                                str1 = str1 +
                                    "<td>&nbsp;" +

```

```

        "</td>";
    }
    }else if( i == count )
    {
        id = resultSet.getString( i ).trim();
        str1 = str1 +
            "<td>" +
            "<form action=\"take.jsp\" method=\"post\">" +
            "<input type=\"hidden\" name=\"firstname\" value=\"" +
resultSet.getString(i-7).trim() + "\">" +
            "<input type=\"hidden\" name=\"middlename\" value=\"" +
resultSet.getString(i-6).trim() + "\">" +
            "<input type=\"hidden\" name=\"lastname\" value=\"" +
resultSet.getString(i-5).trim() + "\">" +
            "<input type=\"hidden\" name=\"street\" value=\"" +
resultSet.getString(i-4).trim() + "\">" +
            "<input type=\"hidden\" name=\"city\" value=\"" +
resultSet.getString(i-3).trim() + "\">" +
            "<input type=\"hidden\" name=\"state\" value=\"" +
resultSet.getString(i-2).trim() + "\">" +
            "<input type=\"hidden\" name=\"zipcode\" value=\"" +
resultSet.getString(i-1).trim() + "\">" +
            "<input type=\"hidden\" name=\"account_id\" value=\"" +
resultSet.getString(i).trim() + "\">" +
            "<input type=\"submit\" value=\"Delete\">" +
            "</form>" +
            "</td>";
    }
    }
    str = str + str1 + "</tr>";
}
}
message = message + str;
}
}catch( SQLException se ){
    se.printStackTrace();
}catch( Exception e ){
    e.printStackTrace();
}
closeConnToDatabase();
}

public String confirm_person()
{
    getConnToDatabase();
    showPerson();
    showPhone();
    closeConnToDatabase();

    return temp1;
}

public void showPerson()
{
    query = "SELECT firstname, middlename, lastname, street, city, state, zipcode from
person" + lineEnd +
            "WHERE account_id='" + telcobean.getAccount_id() + "'; " + lineEnd;
    temp1 = "";
    try{
        no = statement.execute( query );

        if( no )
        {
            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            count = rsmd.getColumnCount();

            while( resultSet.next() )
            {
                String str1 = "<tr><td align=left height=10 width=\"25%\"><font
color=#2F4F4F size=2>";

                for( int i = 1; i <= count; i ++ )
                {
                    if( i == 1 )

```

```

        {
            str1 = str1 + "<b>Name: </b></font></td><td width=\"75%\" .
align=left><font color=#2F4F4F size=2>" + resultSet.getString( i ) + " ";
            }else if( i < 3 ){
                str1= str1 + resultSet.getString( i ) + " ";
            }else if( i == 3 ){
                str1 = str1 + resultSet.getString( i ) + "</font></td></tr>";
            }else if( i == 4 ){
                str1 = str1 + "<tr><td align=left height=10 width=\"25%\"><font
color=#2F4F4F size=2><b>Address: </b></font></td><td width=\"75%\" align=left><font
color=#2F4F4F size=2>" + resultSet.getString( i ) + "</font></td></tr>";
            }else if( i == 5 ){
                str1 = str1 + "<tr><td align=left height=10 width=\"25%\"><font
color=#2F4F4F size=2><b>City: </b></font></td><td width=\"75%\" align=left><font
color=#2F4F4F size=2>" + resultSet.getString( i ) + ", ";
            }else if( i == 7 ){
                str1 = str1 + resultSet.getString( i ) + "</font></td></tr>";
            }else{
                str1 = str1 + resultSet.getString( i ) + " ";
            }
        }
        temp1 = temp1 + str1;
    }
}
}catch( SQLException se ){
    se.printStackTrace();
}catch( Exception e ){
    e.printStackTrace();
}
}

public void showPhone()
{
    boolean found = false;

    query = "SELECT phonenumber from phone " + lineEnd +
        "WHERE person_id='" + telcobean.getAccount_id() + "','";
    temp = "";
    try{
        found = statement.execute( query );

        if( found )
        {
            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            count = rsmd.getColumnCount();

            while( resultSet.next() )
            {
                String str1 = "<tr><td align=left height=10 width=\"25%\"><font
color=#2F4F4F size=2><b>Telephone Number: </b></font></td><td width=\"75%\"
align=left><font color=#2F4F4F size=2>";

                for( int i = 1; i <= count; i ++ )
                {
                    if( i == 1 )
                    {
                        str1 = str1 + resultSet.getString( i ) + "</font></td></tr>";
                    }else{
                        str1 = str1 +
                            "<tr><td align=left height=10 width=\"25%\"><font color=#2F4F4F
size=2><b>Telephone Number: </b></font></td><td width=\"75%\" align=left><font
color=#2F4F4F size=2>" + resultSet.getString( i ) +
                            "</font></td></tr>";
                    }
                }
                temp = temp + str1;
            }
            temp1 = temp1 + temp;
            temp = "";
        }
    }catch( SQLException se ){
        se.printStackTrace();
    }catch( Exception e ){
        e.printStackTrace();
    }
}
}

```

```

    }

    public boolean getYes()
    {
        isyes();
        return yes;
    }

    public boolean getNo()
    {
        isid();
        return no;
    }

    public void delete_person()
    {
        query = "DELETE from person" + lineEnd +
                "WHERE account_id='" + telcobean.getAccount_id().trim() + "'; " +
lineEnd;
        String query1 = "DELETE from phone" + lineEnd +
                "WHERE person_id='" + telcobean.getAccount_id().trim() + "'; " +
lineEnd;
        try{
            count = statement.executeUpdate( query1 );
            count = statement.executeUpdate( query );
        }catch( SQLException ex ){
            ex.printStackTrace();
        }catch( Exception exc ){
            exc.printStackTrace();
        }
    }

    public String getMessage()
    {
        return message;
    }

    public void closeConnToDatabase()
    {
        try{
            resultSet.close();
            statement.close();
        }catch( SQLException se ){
            se.printStackTrace();
        }catch( NullPointerException ne ){
            ne.printStackTrace();
        }

        dbConnection.close();
    }

    public void execute_deletionPerson()
    {
        getConnToDatabase();
        delete_person();
        closeConnToDatabase();
    }
}

```

## Insertion.java

```

// Date: March 11, 2003

package database;

import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.*;
import common.*;

/**
 * Insert new record into database
 */

```



```

public class Insertion
{
    protected String query;
    protected String temp;
    protected String temp1;
    protected String temp2;
    protected String id;
    protected String message;
    protected boolean yes;
    protected boolean no;
    protected boolean yesno;
    protected int count;
    protected int size;
    protected int length;
    protected String lineEnd;

    protected StringTokenizer token;
    protected Statement statement;
    protected ResultSet resultSet;
    protected ResultSetMetaData rsmd;
    protected Connection connection;
    protected DBConnection dbConnection;
    protected TelcoBean telcobean;

    public Insertion()
    {
        this.query = "";
        this.temp = "";
        this.temp1 = "";
        this.temp2 = "";
        this.id = "";
        this.message = "";
        this.yes = false;
        this.no = false;
        this.yesno = false;
        this.count = 0;
        this.size = 0;
        this.length = 0;
        this.lineEnd = System.getProperty( "line.separator" );

        this.statement = null;
        this.resultSet = null;
        this.rsmd = null;
        this.connection = null;
        this.dbConnection = new DBConnection( "telco1" );
    }

    public Insertion( TelcoBean bean )
    {
        this.query = "";
        this.temp = "";
        this.temp1 = "";
        this.temp2 = "";
        this.id = "";
        this.message = "";
        this.yes = false;
        this.no = false;
        this.yesno = false;
        this.count = 0;
        this.size = 0;
        this.length = 0;
        this.lineEnd = System.getProperty( "line.separator" );

        this.statement = null;
        this.resultSet = null;
        this.rsmd = null;
        this.connection = null;
        this.telcobean = bean;
        this.dbConnection = new DBConnection( "telco1" );
    }

    public void getConnToDatabase()
    {
        try{
            connection = dbConnection.getConnection();
            statement = connection.createStatement();
        }
    }
}

```

```

    }catch( SQLException se){
        se.printStackTrace();
    }catch( Exception exc ){
        exc.printStackTrace();
    }
}

public void getCity()
{
    temp = telcobean.getCity().trim();
    token = new StringTokenizer( temp, "," );

    try{
        String city = token.nextToken().trim();
        telcobean.setCity( city );

        String state = token.nextToken().trim();
        telcobean.setState( state );
    }catch( NoSuchElementException e ){
        e.printStackTrace();
    }
}

/** Insert a new record into person table
**/
public void insert_person()
{
    getCity();

    try{
        query = "INSERT INTO person VALUES( " + lineEnd +
            "'" + telcobean.getFirstName().trim() + "', " + lineEnd +
            "'" + telcobean.getMiddleName().trim() + "', " + lineEnd +
            "'" + telcobean.getLastName().trim() + "', " + lineEnd +
            "'" + telcobean.getStreetName().trim() + "', " + lineEnd +
            "'" + telcobean.getCity().trim() + "', " + lineEnd +
            "'" + telcobean.getState().trim() + "', " + lineEnd +
            "'" + telcobean.getZipCode().trim() + "' " + lineEnd +
            " );" + lineEnd;

        statement.executeUpdate( query );
    }catch( SQLException ex ){
        ex.printStackTrace();
    }catch( Exception exc ){
        exc.printStackTrace();
    }
}

/** Insert a new record into phone table
**/
public void insert_phone()
{
    try{
        query = "INSERT INTO phone VALUES( " + lineEnd +
            "'" + telcobean.getAccount_id().trim() + "', " + lineEnd +
            "'" + telcobean.getPhoneNumber().trim() + lineEnd +
            "' );" + lineEnd;

        statement.executeUpdate( query );
    }catch( SQLException ex ){
        ex.printStackTrace();
    }catch( Exception exc ){
        exc.printStackTrace();
    }
}

/** Insert a new record into person table
**/
public void insert()
{
    if( yes )
    {
        if( !no )
        {
            insert_phone();
            message = "<p><font color=#2F4F4F size=2><b>" + telcobean.getPhoneNumber() +
"</b> for<b> " +

```

```

        telcobean.getFirstName() + " " +
        telcobean.getMiddleName() + " " +
        telcobean.getLastName() +
        "</b> has been added to DBMS</font></p>" +
        getRecord();
    }else{
        message = "<p><font color=#2F4F4F size=2>The phone number <b>" +
telcobean.getPhoneNumber() +
        "</b></font>" + " " +
        "<font color=#2F4F4F size=2> is being used by</font></p>" +
        getRecordII();
    }
}
}else{
    if( !no )
    {
        insert_person();
        newId();
        insert_phone();
        message = "<p><font color=#2F4F4F size=2><b>" + telcobean.getFirstName() +
        "<font color=#2F4F4F size=2>" + telcobean.getMiddleName() + "
</font>" + " " +
        "<font color=#2F4F4F size=2>" + telcobean.getLastName() + "</b>
</font>" +
        "<font color=#2F4F4F size=2> has been added to DBMS.</font></p>" +
        showNewRecord();
    }else{
        message = "<p><font color=#2F4F4F size=2>The phone number <b>" +
telcobean.getPhoneNumber() +
        "</b></font>" + " " +
        "<font color=#2F4F4F size=2> is being used by</font></p>" +
        getRecordII();
    }
}
}

/** Search person table,
**  if the specified customer exists, get account_id
**/
public void isyes()
{
    getConnToDatabase();

    query = "SELECT account_id from person" + lineEnd +
        "WHERE street='" + telcobean.getStreetName() + "' " + lineEnd +
        "AND firstname='" + telcobean.getFirstName() + "' " + lineEnd +
        "AND middlename='" + telcobean.getMiddleName() + "' " + lineEnd +
        "AND lastname='" + telcobean.getLastName() + "' " + lineEnd +
        "AND zipcode='" + telcobean.getZipCode() + "' " + lineEnd;

    try{
        yes = statement.execute( query );
        if( yes )
        {
            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            size = rsmd.getColumnCount();
            count = resultSet.getFetchSize();
            yes = false;

            if( count > 0 )
            {
                yes = true;

                while( resultSet.next() )
                {
                    for( int i = 1; i <=size; i ++ )
                    {
                        temp1 = resultSet.getString( i );
                        telcobean.setAccount_id( temp1 );
                    }
                }
            }
        }
    }catch( SQLException se ){
        se.printStackTrace();
    }
}

```

```

    }
    closeConnToDatabase();
}

public void newId()
{
    boolean f = false;

    query = "SELECT account_id from person" + lineEnd +
            "WHERE street='" + telcobebean.getStreetName() + "' " + lineEnd +
            "AND firstname='" + telcobebean.getFirstName() + "' " + lineEnd +
            "AND middlename='" + telcobebean.getMiddleName() + "' " + lineEnd +
            "AND lastname='" + telcobebean.getLastName() + "' " + lineEnd +
            "AND zipcode='" + telcobebean.getZipCode() + "' " + lineEnd;

    try{
        f = statement.execute( query );
        if( f )
        {
            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            size = rsmd.getColumnCount();
            count = resultSet.getFetchSize();

            if( count > 0 )
            {
                while( resultSet.next() )
                {
                    for( int i = 1; i <=size; i ++ )
                    {
                        temp1 = resultSet.getString( i );
                        telcobebean.setAccount_id( temp1 );
                    }
                }
            }
        }
        catch( SQLException se ){
            se.printStackTrace();
        }
    }

    /** Search phone table to see
    ** if being added phone number exists
    **/
    public void exist()
    {
        getConnToDatabase();

        query = "SELECT phonenumber from phone;";

        try{
            no = statement.execute( query );
            if( no )
            {
                resultSet = statement.getResultSet();
                length = resultSet.getFetchSize();
                rsmd = resultSet.getMetaData();
                size = rsmd.getColumnCount();
                no = false;

                if( length > 1 )
                {
                    int i = 0;
                    int j = 0;
                    while( resultSet.next() && j < length )
                    {
                        for( i = 1; i <=size; i ++ )
                        {
                            if( (resultSet.getString( i ).trim()).equals(
telcobebean.getPhoneNumber() ) )
                            {
                                no = true;
                            }
                        }
                        j ++;
                    }
                }
            }
        }
    }

```

```

        }else{
            int i = 0;
            while( resultSet.next() )
            {
                for( i = 1; i <=size; i ++ )
                {
                    if( (resultSet.getString( i ).trim()).equals(
telcobean.getPhoneNumber() ) )
                    {
                        no = true; //yesno = true;
                    }
                }
            }
        }
    }
}
}catch( SQLException se ){
    se.printStackTrace();
}
closeConnToDatabase();
}

public String getId()
{
    getConnToDatabase();

    query = "SELECT * from phone;";
    boolean f = false;

    try{
        f = statement.execute( query );
        if( f )
        {
            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            size = rsmd.getColumnCount();
            f = false;

            int i = 0;
            while( resultSet.next() )
            {
                for( i = 1; i <=size; i ++ )
                {
                    if( (resultSet.getString( i ).trim()).equals(
telcobean.getPhoneNumber().trim() ) )
                    {
                        f = true;
                        id = resultSet.getString( i-1 );
                        break;
                    }
                }
                if( f ) break;
            }
        }
    }catch( SQLException se ){
        se.printStackTrace();
    }
    // closeConnToDatabase();

    return id;
}

public String showNewRecord()
{
    showNewPerson();
    showNewPhone( temp2 );

    return temp1;
}

public void showNewPerson()
{
    query = "SELECT firstname, middlename, lastname, street, city, state, zipcode,
account_id from person" + lineEnd +
"WHERE firstname='" + telcobean.getFirstName() + "' " + lineEnd +
"AND middlename='" + telcobean.getMiddlename() + "' " + lineEnd +
"AND lastname='" + telcobean.getLastName() + "' " + lineEnd +

```

```

        "AND street='" + telcobean.getStreetName() + "' " + lineEnd +
        "AND zipcode='" + telcobean.getZipCode() + "' ";
temp1 = "";
try{
    no = statement.execute( query );

    if( no )
    {
        resultSet = statement.getResultSet();
        rsmd = resultSet.getMetaData();
        count = rsmd.getColumnCount();

        while( resultSet.next() )
        {
            String str1 = "<font color=#2F4F4F size=2>";

            for( int i = 1; i <= count; i ++ )
            {
                if( i == 1 )
                {
                    str1 = str1 + "<b>Name: </b> " + resultSet.getString( i ) + " ";
                }else if( i < 3 ){
                    str1= str1 + resultSet.getString( i ) + " ";
                }else if( i == 3 ){
                    str1 = str1 + resultSet.getString( i ) + "</font><br>";
                }else if( i == 4 ){
                    str1 = str1 + "<font color=#2F4F4F size=2><b>Address: </b>" +
resultSet.getString( i ) + "</font><br>";
                }else if( i == 5 ){
                    str1 = str1 + "<font color=#2F4F4F size=2><b>City: </b>" +
resultSet.getString( i ) + ", ";
                }else if( i == 7 ){
                    str1 = str1 + resultSet.getString( i ) + "</font><br>";
                }else if( i == 8 ){
                    temp2 = resultSet.getString( i );
                }else{
                    str1 = str1 + resultSet.getString( i ) + " ";
                }
            }
            temp1 = temp1 + str1;
        }
    }
} catch( SQLException se ){
    se.printStackTrace();
}

}

public void showNewPhone( String str )
{
    boolean found = false;

    query = "SELECT phonenumber from phone " + lineEnd +
            "WHERE person_id='" + str + "' ";
    temp = "";
    try{
        found = statement.execute( query );

        if( found )
        {
            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            count = rsmd.getColumnCount();

            while( resultSet.next() )
            {
                String str1 = "<font color=#2F4F4F size=2><b>Telephone Number: </b>";

                for( int i = 1; i <= count; i ++ )
                {
                    if( i == 1 )
                    {
                        str1 = str1 + resultSet.getString( i ) + "</font><br>";
                    }else{
                        str1 = str1 +
                            "<font color=#2F4F4F size=2>" + resultSet.getString( i ) +
                            "</font><br>";
                    }
                }
            }
        }
    }
}

```

```

        }
        temp = temp + str1;
    }
    temp1 = temp1 + temp;
    temp = "";
}
} catch( SQLException se ){
    se.printStackTrace();
}
}

public String getRecordII()
{
    String str = getId();

    getConnToDatabase();

    showRecord( str );
    showNewPhone( str );
    str = temp1;
    temp1 = "";

    closeConnToDatabase();

    return str;
}

public String getRecord()
{
    String str = telcobean.getAccount_id();
    showRecord( str );
    showNewPhone( str );
    str = temp1;
    temp1 = "";

    return str;
}

public void showRecord( String str )
{
    query = "SELECT firstname, middlename, lastname, street, city, state, zipcode from
person" + lineEnd +
        "WHERE account_id='" + str + "';";
    temp1 = "";
    try{
        no = statement.execute( query );

        if( no )
        {
            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            count = rsmd.getColumnCount();

            while( resultSet.next() )
            {
                String str1 = "<font color=#2F4F4F size=2>";

                for( int i = 1; i <= count; i ++ )
                {
                    if( i == 1 )
                    {
                        str1 = str1 + "<b>Name: </b>" + resultSet.getString( i ) + " ";
                    } else if( i < 3 )
                    {
                        str1 = str1 + resultSet.getString( i ) + " ";
                    } else if( i == 3 ){
                        str1 = str1 + resultSet.getString( i ) + "</font><br>";
                    } else if( i == 4 ){
                        str1 = str1 + "<font color=#2F4F4F size=2><b>Address: </b>" +
resultSet.getString( i ) + "<br>";
                    } else if( i == 5 ){
                        str1 = str1 + "<font color=#2F4F4F size=2><b>City: </b>" +
resultSet.getString( i ) + ", ";
                    } else if( i == 7 ){
                        str1 = str1 + resultSet.getString( i ) + "</font><br>";
                    }
                }
            }
        }
    }
}

```

```

                }else{
                    str1 = str1 + resultSet.getString( i ) + " ";
                }
            }
            temp1 = temp1 + str1;
        }
    }
} catch( SQLException se ){
    se.printStackTrace();
}
}

public boolean getNo()
{
    exist();

    return no;
}

public boolean getYes()
{
    isyes();

    return yes;
}

public String getMessage()
{
    return message;
}

public void closeConnToDatabase()
{
    try{
        resultSet.close();
        statement.close();
    }catch( SQLException se ){
        se.printStackTrace();
    }catch( NullPointerException ne ){
        ne.printStackTrace();
    }

    dbConnection.close();
}

public void execute_insert()
{
    isyes();
    exist();

    getConnToDatabase();
    insert();
    closeConnToDatabase();
}
}

```

## UpdateName.java

```

// Date: March 11, 2003

package database;

import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.*;
import common.*;

/**
 * Edit names and addresses
 */
public class UpdateName
{

```



```

protected String query;
protected String query1;
protected String temp;
protected String temp1;
protected String temp2;
protected String message;
protected boolean yes;
protected boolean no;
protected boolean yesno;
protected boolean yeap;
protected boolean yeah;
protected boolean edityes;
protected int count;
protected int size;
protected int length;
protected String[] phoneNum;
protected String lineEnd;

protected StringTokenizer token;
protected Statement statement;
protected ResultSet resultSet;
protected ResultSetMetaData rsmd;
protected Connection connection;
protected DBConnection dbConnection;
protected TelcoBean telcobeans;
protected TelcoBean editbean;

public UpdateName()
{
    this.query = "";
    this.query1 = "";
    this.temp = "";
    this.temp1 = "";
    this.temp2 = "";
    this.message = "";
    this.yes = false;
    this.no = false;
    this.yesno = false;
    this.yeap = false;
    this.yeah = false;
    this.edityes = false;
    this.count = 0;
    this.size = 0;
    this.length = 0;
    this.phoneNum = new String[ this.size ];
    this.lineEnd = System.getProperty( "line.separator" );

    this.statement = null;
    this.resultSet = null;
    this.rsmd = null;
    this.connection = null;
    this.dbConnection = new DBConnection( "telcol" );
}

public UpdateName( TelcoBean bean )
{
    this.query = "";
    this.query1 = "";
    this.temp = "";
    this.temp1 = "";
    this.temp2 = "";
    this.message = "";
    this.yes = false;
    this.no = false;
    this.yesno = false;
    this.yeap = false;
    this.yeah = false;
    this.edityes = false;
    this.count = 0;
    this.size = 0;
    this.length = 0;
    this.phoneNum = new String[ this.size ];
    this.lineEnd = System.getProperty( "line.separator" );

    this.statement = null;
    this.resultSet = null;

```

```

        this.rsmd = null;
        this.connection = null;
        this.telcobebean = bean;
        this.dbConnection = new DBConnection( "telco1" );
    }

    public UpdateName( TelcoBean bean, TelcoBean ebean )
    {
        this.query = "";
        this.query1 = "";
        this.temp = "";
        this.temp1 = "";
        this.temp2 = "";
        this.message = "";
        this.yes = false;
        this.no = false;
        this.yesno = false;
        this.yeap = false;
        this.yeah = false;
        this.edityes = false;
        this.count = 0;
        this.size = 0;
        this.length = 0;
        this.phoneNum = new String[ this.size ];
        this.lineEnd = System.getProperty( "line.separator" );

        this.statement = null;
        this.resultSet = null;
        this.rsmd = null;
        this.connection = null;
        this.telcobebean = bean;
        this.editbean = ebean;
        this.dbConnection = new DBConnection( "telco1" );
    }

    public void getConnnToDatabase()
    {
        try{
            connection = dbConnection.getConnection();
            statement = connection.createStatement();
        }catch( SQLException se){
            se.printStackTrace();
        }catch( Exception exc ){
            exc.printStackTrace();
        }
    }

    public void getQuery()
    {
        try{
            query1 = "SELECT person.firstname, person.middlename, person.lastname,
person.street, person.city, " +
            " person.state, person.zipcode, person.account_id from person " + lineEnd +
            "WHERE person.account_id=phone.person_id" + lineEnd +
            "AND person.firstname LIKE '%" + telcobebean.getFirstName().trim() + "%' " +
lineEnd +
            "AND person.middlename LIKE '%" + telcobebean.getMiddleName().trim() + "%' " +
lineEnd +
            "AND person.lastname LIKE '%" + telcobebean.getLastName().trim() + "%' " + lineEnd
+
            "AND phone.phonenumber LIKE '%" + telcobebean.getPhoneNumber().trim() + "%' " +
lineEnd +
            "ORDER BY person.account_id;" + lineEnd;
        }catch( Exception exc ){
            exc.printStackTrace();
        }
    }
    /**
    query1 = "SELECT firstname, middlename," +
            " lastname, street, city," +
            " state, zipcode, account_id" +
            " from person" + lineEnd;

    if( (telcobebean.getFirstName().trim()).equals( "" ) &&
(telcobebean.getMiddleName().trim()).equals( "" ) )
    {
        query1 = query1 +

```

```

        "        WHERE person.lastname='" + telcobean.getLastName().trim() + "' "
+ lineEnd;
    }else if( (telcobean.getFirstName().trim()).equals( " " ) &&
!(telcobean.getMiddleName().trim()).equals( " " ) )
    {
        query1 = query1 +
        "        WHERE person.middlename='" + telcobean.getMiddleName().trim() +
" ' " + lineEnd +
        "        AND person.lastname='" + telcobean.getLastName().trim() + "' " +
lineEnd;
    }else
    {
        query1 = query1 +
        "        WHERE person.firstname='" + telcobean.getFirstName().trim() + "'
" + lineEnd +
        "        AND person.middlename='" + telcobean.getMiddleName().trim() + "'
" + lineEnd +
        "        AND person.lastname='" + telcobean.getLastName().trim() + "' " +
lineEnd;
    }
    query1 = query1 +
        "ORDER BY person.account_id;" + lineEnd;*/
}

public void search_person()
{
    getConnToDatabase();
    getQuery();
    String str = "";
    String id = "";

    try{
        no = statement.execute( query1 );

        if( no )
        {
            message = "";

            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            count = rsmd.getColumnCount();

            for( int i = 1; i <= count; i ++ )
            {
                if( i != count )
                {
                    message = message +
                        "<th><p><font color=#2F4F4F size=2>" + rsmd.getColumnName( i ) +
                        "</p></font></th>";
                }
            }

            message = "<tr>" + message + "</tr>";
            while( resultSet.next() )
            {
                if( !(resultSet.getString( count ).trim()).equals( id ) )
                {
                    str = str + "<tr>";
                    String str1 = "";

                    for( int i = 1; i <= count; i ++ )
                    {
                        if( i != count )
                        {
                            if( !resultSet.getString( i ).equals( " " ) )
                            {
                                str1 = str1 +
                                    "<td><p><font color=#2F4F4F size=2>" + resultSet.getString(
i ) +
                                    "</p></font></td>";
                            }else{
                                str1 = str1 +
                                    "<td>&nbsp;" +
                                    "</td>";
                            }
                        }else if( i == count )

```

```

        {
            id = resultSet.getString( i ).trim();
            str1 = str1 +
                "<td>" +
                "<form action=\"editmenu.jsp\" method=\"post\">" +
                "<input type=\"hidden\" name=\"firstname\" value=\"" +
resultSet.getString(i-7).trim() + "\">" +
                "<input type=\"hidden\" name=\"middlename\" value=\"" +
resultSet.getString(i-6).trim() + "\">" +
                "<input type=\"hidden\" name=\"lastname\" value=\"" +
resultSet.getString(i-5).trim() + "\">" +
                "<input type=\"hidden\" name=\"street\" value=\"" +
resultSet.getString(i-4).trim() + "\">" +
                "<input type=\"hidden\" name=\"city\" value=\"" +
resultSet.getString(i-3).trim() + "\">" +
                "<input type=\"hidden\" name=\"state\" value=\"" +
resultSet.getString(i-2).trim() + "\">" +
                "<input type=\"hidden\" name=\"zipcode\" value=\"" +
resultSet.getString(i-1).trim() + "\">" +
                "<input type=\"hidden\" name=\"account_id\" value=\"" +
resultSet.getString(i).trim() + "\">" +
                "<input type=\"submit\" value=\"Edit\">" +
                "</form>" +
                "</td>";
        }
        str = str + str1 + "</tr>";
    }
}

message = message + str;
}
} catch( SQLException se ){
    se.printStackTrace();
} catch( NullPointerException ne ) {
    ne.printStackTrace();
}
}
closeConnToDatabase();
}

// if customer exists
public void isyes()
{
    getConnToDatabase();
    getQuery();

    try{
        yes = statement.execute( query1 );
        if( yes )
        {
            resultSet = statement.getResultSet();
            count = resultSet.getFetchSize();
            yes = false;

            if( count > 0 )
            {
                yes = true;
            }else{
                yes = false;
            }
        }
    } catch( SQLException se ){
        se.printStackTrace();
    } catch( NullPointerException ne ) {
        ne.printStackTrace();
    }
    closeConnToDatabase();
}

public boolean getYes()
{
    isyes();
    return yes;
}

public void getQuery1()

```

```

    {
        query = "SELECT firstname, middlename," +
            " lastname, street, city," +
            " state, zipcode" +
            " from person" + lineEnd;

        query = query +
            " WHERE firstname='" + telcobean.getFirstName().trim() + "' " +
lineEnd +
            " AND middlename='" + telcobean.getMiddleName().trim() + "' " +
lineEnd +
            " AND lastname='" + telcobean.getLastName().trim() + "' " + lineEnd
+
            " AND street='" + telcobean.getStreetName().trim() + "' " + lineEnd
+
            " AND zipcode='" + telcobean.getZipCode().trim() + "' " + lineEnd +
            " AND account_id='" + telcobean.getAccount_id().trim() + "';" +
lineEnd;
    }

    public void getQuery2()
    {
        query = "SELECT firstname, middlename," +
            " lastname, street, city," +
            " state, zipcode" +
            " from person" + lineEnd;

        query = query +
            " WHERE account_id='" + telcobean.getAccount_id().trim() + "';" +
lineEnd;
    }

    public void getQuery5()
    {
        query = "SELECT phonenumber from phone" + lineEnd;

        query = query +
            " WHERE person_id='" + telcobean.getAccount_id().trim() + "';" +
lineEnd;
    }

    public TelcoBean getEditbean()
    {
        return editbean;
    }

    public String current()
    {
        String str = "";
        getConnToDatabase();
        current_person();
        getQuery5();
        current_phone();
        closeConnToDatabase();

        str = temp1;
        temp1 = "";

        return str;
    }

    public String editnew()
    {
        String str = "";
        getConnToDatabase();
        getQuery2();
        new_person();
        getQuery5();
        current_phone();
        closeConnToDatabase();

        str = temp1;
        temp1 = "";

        return str;
    }
}

```

```

public void current_person()
{
    getQuery1();
    boolean fo = false;

    try{
        fo = statement.execute( query );

        if( fo )
        {
            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            count = rsmd.getColumnCount();

            while( resultSet.next() )
            {
                String str1 = "<tr><td align=left height=10 width=\"25%\"><font
color=#2F4F4F size=2>";

                for( int i = 1; i <= count; i ++ )
                {
                    if( i == 1 )
                    {
                        str1 = str1 + "<b>Name: </b></font></td><td width=\"75%\"
align=left><font color=#2F4F4F size=2>" + resultSet.getString( i ) + " ";
                    }else if( i < 3 ){
                        str1 = str1 + resultSet.getString( i ) + " ";
                    }else if( i == 3 ){
                        str1 = str1 + resultSet.getString( i ) + "</font></td></tr>";
                    }else if( i == 4 ){
                        str1 = str1 + "<tr><td align=left height=10 width=\"25%\"><font
color=#2F4F4F size=2><b>Address: </b></font></td><td width=\"75%\" align=left><font
color=#2F4F4F size=2>" + resultSet.getString( i ) + "</font></td></tr>";
                    }else if( i == 5 ){
                        str1 = str1 + "<tr><td align=left height=10 width=\"25%\"><font
color=#2F4F4F size=2><b>City: </b></font></td><td width=\"75%\" align=left><font
color=#2F4F4F size=2>" + resultSet.getString( i ) + ", ";
                    }else if( i == 7 ){
                        str1 = str1 + resultSet.getString( i ) + "</font></td></tr>";
                    }else{
                        str1 = str1 + resultSet.getString( i ) + " ";
                    }
                }
                templ = templ + str1;
            }
        }
    }catch( SQLException se ){
        se.printStackTrace();
    }catch( NullPointerException ne ) {
        ne.printStackTrace();
    }
}

public void new_person()
{
    boolean fo = false;

    try{
        fo = statement.execute( query );

        if( fo )
        {
            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            count = rsmd.getColumnCount();

            while( resultSet.next() )
            {
                String str1 = "<tr><td align=left height=10 width=\"25%\"><font
color=#2F4F4F size=2>";

                for( int i = 1; i <= count; i ++ )
                {
                    if( i == 1 )
                    {

```

```

        str1 = str1 + "<b>Name: </b></font></td><td width=\"75%\"
align=left><font color=#2F4F4F size=2>" + resultSet.getString( i ) + " ";
        }else if( i < 3 ){
            str1= str1 + resultSet.getString( i ) + " ";
        }else if( i == 3 ){
            str1 = str1 + resultSet.getString( i ) + "</font></td></tr>";
        }else if( i == 4 ){
            str1 = str1 + "<tr><td align=left height=10 width=\"25%\"><font
color=#2F4F4F size=2><b>Address: </b></font></td><td width=\"75%\" align=left><font
color=#2F4F4F size=2>" + resultSet.getString( i ) + "</font></td></tr>";
        }else if( i == 5 ){
            str1 = str1 + "<tr><td align=left height=10 width=\"25%\"><font
color=#2F4F4F size=2><b>City: </b></font></td><td width=\"75%\" align=left><font
color=#2F4F4F size=2>" + resultSet.getString( i ) + ", ";
        }else if( i == 7 ){
            str1 = str1 + resultSet.getString( i ) + "</font></td></tr>";
        }else{
            str1 = str1 + resultSet.getString( i ) + " ";
        }
    }
    temp1 = temp1 + str1;
}
}
} catch( SQLException se ){
    se.printStackTrace();
} catch( NullPointerException ne ) {
    ne.printStackTrace();
}
}

public void current_phone()
{
    boolean found = false;
    temp = "";
    try{
        found = statement.execute( query );

        if( found )
        {
            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            count = rsmd.getColumnCount();

            while( resultSet.next() )
            {
                String str1 = "<tr><td align=left height=10 width=\"25%\"><font
color=#2F4F4F size=2><b>Telephone Number: </b></font></td><td width=\"75%\"
align=left><font color=#2F4F4F size=2>";

                for( int i = 1; i <= count; i ++ )
                {
                    if( i == 1 )
                    {
                        str1 = str1 + resultSet.getString( i ) + "</font></td></tr>";
                    }else{
                        str1 = str1 +
                            "<tr><td align=left height=10 width=\"25%\"><font color=#2F4F4F
size=2><b>Telephone Number: </b></font></td><td width=\"75%\" align=left><font
color=#2F4F4F size=2>" + resultSet.getString( i ) +
                            "</font></td></tr>";
                    }
                }
                temp = temp + str1;
            }
            temp1 = temp1 + temp;
            temp = "";
        }
    } catch( SQLException se ){
        se.printStackTrace();
    } catch( NullPointerException ne ) {
        ne.printStackTrace();
    }
}

// if the customer uses is using this phone number

```

```

public void isno()
{
    getConnToDatabase();

    try{
        getQuery1();
        no = statement.execute( query );
        if( no )
        {
            resultSet = statement.getResultSet();
            count = resultSet.getFetchSize();
            no = false;

            if( count > 0 )
            {
                no = true;
            }else{
                no = false;
            }
        }
    }catch( SQLException se ){
        se.printStackTrace();
    }catch( NullPointerException ne ) {
        ne.printStackTrace();
    }
    closeConnToDatabase();
}

public void editName()
{
    if( no )
    {
        if( !(editbean.getFirstName().trim()).equals( (telcobean.getFirstName().trim())
) &&
        !(editbean.getMiddleName().trim()).equals( (telcobean.getMiddleName().trim())
) &&
        !(editbean.getLastName().trim()).equals( (telcobean.getLastName().trim()) ) )
        {
            query = "UPDATE person " + lineEnd +
                "SET firstName = '" + editbean.getFirstName().trim() + "', " +
                "middleName = '" + editbean.getMiddleName().trim() + "', " +
                "lastName= '" + editbean.getLastName().trim() + "' " + lineEnd +
                "WHERE account_id='" + telcobean.getAccount_id() + "';" + lineEnd;
        }else if( !(editbean.getFirstName().trim()).equals(
(telcobean.getFirstName().trim()) ) &&
        !(editbean.getLastName().trim()).equals(
(telcobean.getLastName().trim()) ) )
        {
            query = "UPDATE person " + lineEnd +
                "SET firstName = '" + editbean.getFirstName().trim() + "', " +
                "lastName= '" + editbean.getLastName().trim() + "' " + lineEnd +
                "WHERE account_id='" + telcobean.getAccount_id() + "';" + lineEnd;
        }else if( !(editbean.getFirstName().trim()).equals(
(telcobean.getFirstName().trim()) ) &&
        !(editbean.getMiddleName().trim()).equals(
(telcobean.getMiddleName().trim()) ) )
        {
            query = "UPDATE person " + lineEnd +
                "SET firstName = '" + editbean.getFirstName().trim() + "', " +
                "middleName= '" + editbean.getMiddleName().trim() + "' " + lineEnd +
                "WHERE account_id='" + telcobean.getAccount_id() + "';" + lineEnd;
        }else if( !(editbean.getMiddleName().trim()).equals(
(telcobean.getMiddleName().trim()) ) &&
        !(editbean.getLastName().trim()).equals(
(telcobean.getLastName().trim()) ) )
        {
            query = "UPDATE person " + lineEnd +
                "SET middleName = '" + editbean.getMiddleName().trim() + "', " +
                "lastName= '" + editbean.getLastName().trim() + "' " + lineEnd +
                "WHERE account_id='" + telcobean.getAccount_id() + "';" + lineEnd;
        }else if( !(editbean.getFirstName().trim()).equals(
(telcobean.getFirstName().trim()) ) )
        {
            query = "UPDATE person " + lineEnd +
                "SET firstName = '" + editbean.getFirstName().trim() + "' " + lineEnd +
                "WHERE account_id='" + telcobean.getAccount_id() + "';" + lineEnd;
        }
    }
}

```



```

        }else if( !(editbean.getMiddleName().trim()).equals(
(telcobebean.getMiddleName().trim()) ) )
        {
            query = "UPDATE person " + lineEnd +
                    "SET middleName = '" + editbean.getMiddleName().trim() + "' " + lineEnd
+
                    "WHERE account_id='" + telcobebean.getAccount_id() + "';" + lineEnd;
        }else if( !(editbean.getLastName().trim()).equals(
(telcobebean.getLastName().trim()) ) )
        {
            query = "UPDATE person " + lineEnd +
                    "SET lastName= '" + editbean.getLastName().trim() + "' " + lineEnd +
                    "WHERE account_id='" + telcobebean.getAccount_id() + "';" + lineEnd;
        }else{
            message = "<p><font color=#2F4F4F size=2> Update failed.</font></p><br>";
        }

        getConnToDatabase();
        try{
            size= statement.executeUpdate( query );

            if( size != 0 )
            {
                message = "<div align=center><table border=0 width=\"100%\"> " +
                        "<tr><td align=left> " +
                        "<p><font size=3 color=#2F4F4F>The current record</font></p> " +
                        "</td></tr>" +
                        editnew() +
                        "</table></div>";
            }
        }catch( SQLException se){
            se.printStackTrace();
        }catch( NullPointerException ne ) {
            ne.printStackTrace();
        }
    }else{
        message = "<font color=#2F4F4F size=2>" + telcobebean.getFirstName().trim() +
"</font>" + " " +
"<font color=#2F4F4F size=2>" + telcobebean.getMiddleName().trim() +
"</font>" + " " +
"<font color=#2F4F4F size=2>" + telcobebean.getLastName().trim() +
"</font>" +
"<p><font color=#2F4F4F size=2> is not in database
system.</font></p><br>";
    }
}

public void editAddress()
{
    if( no )
    {
        if( !(editbean.getStreetName().trim()).equals( telcobebean.getStreetName().trim()
) &&
            !(editbean.getZipCode().trim()).equals( telcobebean.getZipCode().trim() ) )
        {
            query = "UPDATE person " + lineEnd +
                    "SET street= '" + editbean.getStreetName().trim() + "', " +
                    "zipCode= '" + editbean.getZipCode().trim() + "' " + lineEnd +
                    "WHERE account_id='" + telcobebean.getAccount_id() + "';" + lineEnd;
        }else if( !(editbean.getStreetName().trim()).equals(
telcobebean.getStreetName().trim() ) )
        {
            query = "UPDATE person " + lineEnd +
                    "SET street= '" + editbean.getStreetName().trim() + "' " + lineEnd +
                    "WHERE account_id='" + telcobebean.getAccount_id() + "';" + lineEnd;
        }else if( !(editbean.getZipCode().trim()).equals( telcobebean.getZipCode().trim()
) )
        {
            query = "UPDATE person " + lineEnd +
                    "SET zipCode= '" + editbean.getZipCode().trim() + "' " + lineEnd +
                    "WHERE account_id='" + telcobebean.getAccount_id() + "';" + lineEnd;
        }else{
            message = "<p><font color=#2F4F4F size=2> Update failed. Try
Again.</font></p><br>";
        }
    }
}

```

```

        try{
            size= statement.executeUpdate( query );
            if( size != 0 )
            {
                message = "<div align=center><table border=0 width=\"100%\">\" +
                    "<tr><td align=left>\" +
                    "<p><font size=3 color=#2F4F4F>The current record</font></p>\" +
                    "</td></tr>\" +
                    editnew() +
                    "</table></div>\";
            }
        }catch( SQLException se){
            se.printStackTrace();
        }catch( Exception e ){
            e.printStackTrace();
        }
    }else{
        message = "<font color=#2F4F4F size=2>\" + telcobean.getFirstName().trim() +
        "</font>\" + \" \" +
        "<font color=#2F4F4F size=2>\" + telcobean.getMiddleName().trim() +
        "</font>\" + \" \" +
        "<font color=#2F4F4F size=2>\" + telcobean.getLastName().trim() +
        "</font>\" +
        "<p><font color=#2F4F4F size=2> is not in database
        system.</font></p><br>\";
    }
}

// if the customer exists
public boolean getNo()
{
    isno();
    return no;
}

public String getMessage()
{
    return message;
}

public void closeConnToDatabase()
{
    try{
        resultSet.close();
        statement.close();
    }catch( SQLException se ){
        se.printStackTrace();
    }catch( NullPointerException ne ){
        ne.printStackTrace();
    }
}

dbConnection.close();
}

public void execute_editName()
{
    isno();
    getConnToDatabase();
    editName();
    closeConnToDatabase();
}

public void execute_editAddress()
{
    isno();
    getConnToDatabase();
    editAddress();
    closeConnToDatabase();
}
}

```

## Search.java

// Date: March 11, 2003

```
package database;

import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.*;
import common.*;

/**
 * Search records
 */
public class Search
{
    protected String query;
    protected String temp;
    protected String temp1;
    protected String message;
    protected boolean yes;
    protected boolean no;
    protected boolean yesno;
    protected int count;
    protected int size;
    protected int length;
    protected String lineEnd;

    protected Statement statement;
    protected ResultSet resultSet;
    protected ResultSetMetaData rsmd;
    protected Connection connection;
    protected DBConnection dbConnection;
    protected TelcoBean bean;

    public Search( TelcoBean bean )
    {
        this.query = "";
        this.temp = "";
        this.temp1 = "";
        this.message = "";
        this.yes = false;
        this.no = false;
        this.yesno = false;
        this.count = 0;
        this.size = 0;
        this.length = 0;
        this.lineEnd = System.getProperty( "line.separator" );

        this.statement = null;
        this.resultSet = null;
        this.rsmd = null;
        this.connection = null;
        this.bean = bean;
        this.dbConnection = new DBConnection( "telco1" );
    }

    public void getConnToDatabase()
    {
        try{
            connection = dbConnection.getConnection();
            statement = connection.createStatement();
        }catch( SQLException se){
            se.printStackTrace();
        }catch( Exception exc ){
            exc.printStackTrace();
        }
    }

    public void getQuery1()
    {
        try{
            query = "SELECT person.firstname, person.middlename, person.lastname,
person.street, person.city, " +

```

```

        " person.state, person.zipcode, phone.phonenumber, person.account_id from
person, phone " + lineEnd +
        "WHERE person.account_id=phone.person_id" + lineEnd +
        "AND person.firstname LIKE '%" + bean.getFirstName().trim() + "%' " + lineEnd +
        "AND person.middlename LIKE '%" + bean.getMiddleName().trim() + "%' " + lineEnd
+
        "AND person.lastname LIKE '%" + bean.getLastName().trim() + "%' " + lineEnd +
+
        "AND phone.phonenumber LIKE '%" + bean.getPhoneNumber().trim() + "%' " + lineEnd
+
        "ORDER BY person.account_id;" + lineEnd;
    }catch( Exception exc ){
        exc.printStackTrace();
    }
}

public void isyes()
{
    getConnToDatabase();
    getQuery1();

    try{
        yes = statement.execute( query );
        if( yes )
        {
            resultSet = statement.getResultSet();
            count = resultSet.getFetchSize();
            yes = false;

            if( count > 0 )
            {
                yes = true;
            }else{
                yes = false;
            }
        }
    }catch( SQLException se ){
        se.printStackTrace();
    }catch( NullPointerException ne ) {
        ne.printStackTrace();
    }
    closeConnToDatabase();
}

public boolean getYes()
{
    isyes();
    return yes;
}

public void review_person()
{
    String str = "";

    getQuery1();
    try{
        no = statement.execute( query );

        if( no )
        {
            message = "";

            resultSet = statement.getResultSet();
            rsmd = resultSet.getMetaData();
            count = rsmd.getColumnCount();

            for( int i = 1; i <= count; i ++ )
            {
                message = message +
                    "<th><p><font color=#2F4F4F size=2>" + rsmd.getColumnName( i ) +
                    "</p></font></th>";
            }
            message = "<tr>" + message + "</tr>";

            while( resultSet.next() )
            {
                str = str + "<tr>";
            }
        }
    }
}

```

```

        String str1 = "";
        for( int i = 1; i <= count; i ++ )
        {
            if( !resultSet.getString( i ).equals( "" ) )
            {
                str1 = str1 +
                    "<td><p><font color=#2F4F4F size=2>" + resultSet.getString( i
) +
                    "</p></font></td>";
            }else{
                str1 = str1 +
                    "<td>&nbsp;" +
                    "</td>";
            }
            str = str + str1 + "</tr>";
        }

        message = message + str;
    }
    }catch( SQLException se ){
        se.printStackTrace();
    }
}

public String getMessage()
{
    return message;
}

public void closeConnToDatabase()
{
    try{
        resultSet.close();
        statement.close();
    }catch( SQLException se ){
        se.printStackTrace();
    }catch( NullPointerException ne ){
        ne.printStackTrace();
    }

    dbConnection.close();
}

public void search( )
{
    getConnToDatabase();
    review_person();
    closeConnToDatabase();
}
}

```

## REFERENCES

- [1]. J. Gettys, et al. "Hypertext Transfer Protocol-HTTP/1.1", The Internet Society, June 1999  
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- [2]. Don Box, et al, "Simple Object Access Protocol (SOAP) 1.1", W3C Note 08 May 2000  
<http://www.w3.org/TR/SOAP/>
- [3]. T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation 6 October 2000.  
<http://www.w3.org/TR/REC-xml>
- [4]. E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, "Web Services Description Language (WSDL) 1.1", W3C Note 15 March 2001.  
<http://www.w3.org/TR/wsdl>
- [5]. T. Bellwood, et al, "UDDI Version 3.0, Published Specification 19 July 2002.  
<http://uddi.org/pubs/uddi-v3.00-published-20020719.pdf>
- [6]. J. Snell, D. Tidwell, P. Kulchenko, Programming Web Services with SOAP.

[7]. Scott Ambler, "An Ounce of Prevention: Class Type Architectures", Software Development Magazine (Mar 1996) pp. 4045-61.

[8]. W B Noffsinger & Rober Niedbalski & Michael Blanks & Niall Emmart, "Legacy Object Modelling Speeds Software Integration", Commun ACM V41n12 (Dec 1998) pp80-89.

[9]. Jeff Offutt, "Quality Attributes of Web Software Applications", IEEE Software Magazine V19n2 (Mar/Apr 2002) pp25-32.

[10]. Arindam Banerji, etal. "Web Service Conversation Language", W3C Note 14 March 2002.  
<http://www.w3.org/TR/wscl10/>